การแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีโดยอัตโนมัติสำหรับการทวนสอบเชิงรูปนัย

นางสาวชลิกา ศักดิ์สุภาวัฒนกุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2560

Automatic Transformation of Timed Petri Nets into Event-B for Formal Verification

Miss Chalika Saksupawattanakul

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Software Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2017

หัวข้อวิทยานิพนธ์	การแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีโดยอัตโนมัติสำหรับการทวนสอบเชิงรูปนัย
โดย	นางสาวชลิกา ศักดิ์สุภาวัฒนกุล
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

.....กรรมการ

(รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี)

.....กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร.สมศักดิ์ วาณิชอนันต์ชัย)

ชลิกา ศักดิ์สุภาวัฒนกุล : การแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีโดยอัตโนมัติสำหรับการทวนสอบเชิงรูปนัย (Automatic Transformation of Timed Petri Nets into Event-B for Formal Verification) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.วิวัฒน์ วัฒนาวุฒิ, 119 หน้า.

โดยทั่วไปการออกแบบซอฟต์แวร์เป็นขั้นตอนที่สำคัญในกระบวนการพัฒนาซอฟต์แวร์ ความถูกต้องของพฤติกรรมของระบบซอฟต์แวร์แบบเรียลไทม์มีความสัมพันธ์กับเวลาของการทำงานของระบบ หากพบข้อผิดพลาดหลังจากการพัฒนาซอฟต์แวร์จะทำให้มีผลกระทบที่ต้องสูญเสียอย่างมาก จึงเป็นสิ่งสำคัญที่ต้องทำการทวนสอบแบบจำลองของการออกแบบ เพื่อหาจุดผิดพลาดก่อนที่จะลงเมื่อพัฒนาซอฟต์แวร์ วิทยานิพนธ์นี้จึงนำเสนอทางเลือกในการทวนสอบเชิงรูปนัยโดยใช้ไทมด์เพทริเน็ตในการสร้างแบบจำลองเชิงรูปนัย ซึ่งในปฏิบัติแล้วไทมด์เพทริเน็ตเป็นวิธีการเชิงรูปนัยที่ใช้สัญลักษณ์กราฟิกสำหรับจำลองโครงสร้างของระบบ ซึ่งทำให้สามารถเข้าใจพฤติกรรมของระบบได้ง่าย แต่ไทมด์เพทริเน็ตยังขาดการแสดงส่วนของข้อมูลที่ใช้ภายในระบบและยังไม่ได้มีการปรับแต่งใดๆ หากระบบมีขนาดใหญ่และมีความซับซ้อนมาก อาจจะเกิดปัญหาการระบบของสถานะได้ State explosion อีกทางเลือกของวิธีการเชิงรูปนัยพบว่าอีเวนท์บี เป็นวิธีการเชิงรูปนัยที่นิยมสำหรับการทวนสอบการทำงานของระบบโดยสนใจข้อมูลภายในระบบอีกทั้งสนับสนุนการปรับแต่งของระบบซอฟต์แวร์แบบเรียลไทม์ โดยใช้วิธีการพิสูจน์ทฤษฎีทางคณิตศาสตร์ ซึ่งจะทำให้ไม่เกิดปัญหาการระเบิดของสถานะในระหว่างการทวนสอบ อย่างไรก็ตามการเขียนอีเวนท์บีไม่ง่ายนักเนื่องจากต้องมีทักษะทางด้านคณิตศาสตร์สำหรับการเขียนอีเวนท์บีเพื่อทวนสอบระบบ

งานวิทยานิพนธ์นี้จึงเสนอเครื่องมือการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีโดยอัตโนมัติสำหรับการทวนสอบเชิงรูปนัย โดยสนใจแบบจำลองไทมด์เพทริเน็ตที่มีค่าน้ำหนักโทเค็นที่มีค่าเท่ากับ 1 เท่านั้น และกฎการแปลงส่วนประกอบไทมด์เพทริเน็ตให้เป็นอีเวนท์บีทั้งหมด 7 ข้อ ข้อมูลนำเข้าเครื่องมือการแปลงจะเป็นแบบจำลองไทมด์เพทริเน็ตที่อยู่ในรูปแบบแฟ้มเอกสารเอกซ์เอ็มแอล และเครื่องมือจะดำเนินการแปลงโดยใช้กฎการแปลงที่ได้นิยามขึ้นมา เพื่อแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บีได้โดยอัตโนมัติ เมื่อได้ผลลัพธ์การแปลงอีเวนท์บีเรียบร้อยแล้ว จะดำเนินการทวนสอบด้วยเครื่องมือโรดิน เพื่อตรวจสอบความถูกต้องของวิธีการแปลงโดยใช้เครื่องมือการแปลงและทวนสอบการทำงานของระบบ

5970914321 : MAJOR SOFTWARE ENGINEERING

KEYWORDS: TIMED PETRI NET / EVENT-B / FORMAL VERIFICATION / REAL-TIME SOFTWARE SYSTEM

CHALIKA SAKSUPAWATTANAKUL: Automatic Transformation of Timed Petri Nets into Event-B for Formal Verification. ADVISOR: ASSOC. PROF. WIWAT VATANAWOOD, Ph.D., 119 pp.

Software design is typically an important step in the software development process. The correctness of the behavioral of real-time software system relies on both the result of its computation and the clock times. If errors are found after the software development process, it will definitely cause significant loss to the software developer. To prevent such loss and reduce development effort, it is practically crucial to verify the design model and find errors before the development software process starts. This study, we focus on formal methods between two well-known verification methods: (i) timed Petri net and (ii) Event-B. Timed Petri net method can be used to create verification model by leveraging a graphical framework to make ease for understanding in term of behavior for software. However, it still lacks the illustration of the internal data and refinement process. If the system is effectively large and complicated, the corresponding verification model can introduce a problem of the state explosion. Alternatively, formal verification by using Event-B specification method provides an efficient automatic theorem proving tool by using mathematical logic proving that will not cause the problem of state explosion. However, writing an Event-B specification, to verify software, from scratch is non-trivial and requires a strong background in mathematics.

This thesis proposes an automatic tool for transformation from timed Petri net into Event-B for formal verification. This thesis focuses on timed Petri net model that has the weight of token as one. Seven mapping rules required to transform timed Petri net into Event-B. The input of our automatic tool supports timed Petri net model described in XML format. The tool applies mapping rules and automatically transforms timed Petri net into Event-B. We use Rodin tool to verify the correctness of the generated Event-B specification and to further verify the behavior of the corresponding system.

Department: Computer Engineering Student's Signature .....

Field of Study: Software Engineering Advisor's Signature .....

Academic Year: 2017

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุลวงไปด้วยดีจากคำแนะนำของท่าน รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้สละเวลาอันมีค่าเพื่อให้คำปรึกษาและแนวทางในการทำงานวิจัยที่ดีและคอยชี้แนะวิธีการแก้ปัญหาในแต่ละด้านเพื่อให้เกิดผลสัมฤทธิ์สูงสุดและเป็นประโยชน์ต่อการทำวิจัยและสามารถนำไปถ่ายทอดแก่ส่วนรวมได้โดยสมบูรณ์ ผู้วิจัยจึงขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

ขอขอบพระคุณท่าน ศาสตราจารย์ ดร. ประภาส จงสถิตย์วัฒนา ประธานกรรมการสอบวิทยานิพนธ์ รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรีและผู้ช่วยศาสตราจารย์ ดร.สมศักดิ์ วาณิชอนันต์ชัย กรรมการสอบวิทยานิพนธ์ ที่ได้กรุณาเสียสละเวลาอันมีค่ายิ่ง เพื่อให้ข้อแนะนำและแนวทางการดำเนินการ ตลอดจนความรู้และแง่คิดที่เป็นประโยชน์ต่อการทำวิจัย ซึ่งทำให้วิทยานิพนธ์สำเร็จลุล่วงไปด้วยดี

ขอขอบพระคุณคณาจารย์ ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ที่คอยอบรมสั่งสอนและประสิทธิ์ประสาทวิชาความรู้ตอลกจนถ่ายทอดประสบการณ์อันมีค่า

สุดท้ายขอกราบขอบพระคุณบิดา มารดาและครอบครัว ที่คอยให้กำลังใจ และแนวคิดสำหรับการดำรงชีวิตที่ดีและสนับสนุนทั้งเรื่องงานและการศึกษาด้วยดีเสมอมา

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง 1
สารบัญรูป 4
บทที่ 1 บทนำ 11
1.1 ที่มาและความสำคัญของปัญหา 11
1.2 วัตถุประสงค์ 12
1.3 ขอบเขตการดำเนินงาน 12
1.4 ขั้นตอนการดำเนินงาน 13
1.5 ประโยชน์ที่คาดว่าจะได้รับ 14
1.6 บทความวิจัยที่ได้รับการตีพิมพ์ 14
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 15
2.1 ทฤษฎีที่เกี่ยวข้อง 15
2.1.1 เพทริเน็ต (Petri Nets) 15
2.1.1.1 รูปแบบของเพทริเน็ต (Pattern of Petri Nets) 17
2.1.1.2 พฤติกรรมของเพทริเน็ต (Behavior of Petri Nets) 18
1) เปิดใช้งานทรานซิชัน (Enabled Transitions) 18
2) ยิงทรานซิชัน (Firing Transitions) 18
2.1.1.3 ไทมด์เพทริเน็ต (Timed Petri Nets) 18
2.1.1.4 พฤติกรรมของไทมด์เพทริเน็ต (Behavior of timed Petri net) 19

หน้า

1) เปิดใช้งานทรานซิชัน (Enabled Transitions) 19
2) ยิงทรานซิชัน (Firing Transitions) 19
2.1.2 อีเวนท์บี (Event-B) 20
2.1.2.1 ส่วนประกอบของอีเวนท์บี (Component of Event-B) 20
1) บริบท (Context) 21
2) แมชชีน (Machine) 21
3) อีเวนท์ (Events) 22
4) ความสัมพันธ์ระหว่างบริบทและแมชชีน (Relationship between context and machine) 23
5) สัญกรณ์ทางคณิตศาสตร์ของอีเวนท์บี (Notation of mathematics in Event-B) 24
2.1.2.2 เครื่องมือโรดิน (Rodin tool) 24
2.1.2.3 การพิสูจน์ (Proving) 25
1) การตรวจสอบแบบสถิต (Static check) 25
2) ตัวสร้างข้อผูกพัน (Proof obligation generator) 26
3) ผู้พิสูจน์ (Prover) 26
4) ส่วนต่อขยายเอสเอ็มที (SMT plug in) 27
5) ส่วนต่อขยายโปรบี (ProB plug in) 28
2.1.2.4 ตรรกศาสตร์เชิงกาลเวลา (Temporal Logic/ Liner Temporal Logic) 29
1) ไวยากรณ์และความหมาย (Syntax and Semantic) 29
2) คุณสมบัติของตรรกะเชิงเวลา (Linear Temporal properties) 30
2.2 งานวิจัยที่เกี่ยวข้อง 30

2.2.1 งานวิจัยชื่อ Semantic Embedding of Petri Nets into Event-B โดย Christian Attiogbe ปี ค.ศ. 2005 30

2.2.2 งานวิจัยชื่อ The EventB2PN Tool: From Event-B Specification to Petri Nets through Model Transformation โดย Mohamed Garoui, Belhassen Mazigh และ Abderrafiaa Koukam ปี ค.ศ. 2015 31

2.2.3 งานวิจัยชื่อ Petri Nets to B-Language Transformation in Software Development โดย Stefan Korecko และ Branislav Sobota ปี ค.ศ. 2014..... 32

2.2.4 งานวิจัยชื่อ From TiMo to Event-B Event-Driven Timed Mobility โดย Gabriel Ciobanu Thai Son Hoang Alin Stefanescu ปี ค.ศ. 2014 32

2.2.5 งานวิจัยชื่อ Modeling Critical Systems with Timing Constraints in Event-B โดย Siavashi Faezeh, Wald Marina และ Tsiopoulos Leonidas ปี ค.ศ. 2013 33

บทที่ 3 การแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี 34

3.1 แนวคิดการแปลงแบบจำลองไทมด์เพทริเน็ตในรูปแบบแฟ้มเอกสารเอกซ์เอ็มแอล 35

3.1.1 การสกัดและตรวจสอบส่วนประกอบของแบบจำลองไทมด์เพทริเน็ต 35

3.1.2 การนิยามกฎการแปลงส่วนประกอบของแบบจำลองไทมด์เพทริเน็ตให้อยู่ในรูปแบบของเอกซ์เอ็มแอล 35

1) การนิยามป้ายระบุเปิดและป้ายระบุปิด 36

2) การนิยามรากของส่วนย่อย 36

3) การออกแบบเค้าร่างเอกซ์เอ็มแอลสำหรับแบบจำลองไทมด์เพทริเน็ต 36

4) การนิยามเค้าร่างไทมด์เพทริเน็ตให้อยู่ในรูปแบบแฟ้มเอกซ์เอ็มแอล 37

3.2 การแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บี 39

3.2.1 การนิยามกฎการแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บี 39

หน้า

กฎข้อที่ 1 กฎการแปลงส่วนประกอบเพลส 39
กฎข้อที่ 2 กฎการแปลงส่วนประกอบทรานซิชัน 40
กฎข้อที่ 3 กฎการแปลงส่วนประกอบของการไหลของเส้นทางการย้ายสถานะ 41
กฎข้อที่ 4 กฎการแปลงช่วงเวลาของแต่ละทรานซิชัน 41
กฎข้อที่ 5 กฎการนิยามตัวแปร (Variables) และตัวยืนยง (Invariants) ในอีเวนท์บี 42
กฎข้อที่ 6 กฎการแปลงเหตุการณ์เริ่มต้น (Initialisation Events) 43
กฎข้อที่ 7 กฎการแปลงเหตุการณ์แบบไดนามิก (Dynamic Events) 44
3.3 รูปแบบการนิยามสัญลักษณ์ทางคณิตศาสตร์สำหรับแสดงในผลลัพธ์การแปลงอีเวนท์บี 51
ตัวอย่างการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี 52
บทที่ 4 การออกแบบและการพัฒนาเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติ 56
4.1 การวิเคราะห์และออกแบบเครื่องมือ 56
4.1.1 แผนภาพยูสเคส 56
4.2 การออกแบบภาพรวมการทำงานของเครื่องมือการแปลงไทมด์เพริเน็ตให้เป็นอีเวนท์ 65
4.2.1 การออกแบบส่วนต่อประสานผู้ใช้งาน 67
1) หน้าจอนำเข้าจำลองไทมด์เพทริเน็ตที่อยู่ในรูปแบบของแฟ้มเอกสารเอกซ์เอ็มแอล 67
2) หน้าจอสำหรับหาที่อยู่ของแฟ้มเอกสารนำเข้าเอกซ์เอ็มแอล 67
3) หน้าจอแสดงส่วนประกอบที่ได้จากแฟ้มเอกสารนำเข้าเอกซ์เอ็มแอล 68
4) หน้าจอสำหรับเลือกที่อยู่ของผลลัพธ์การแปลง 68
5) หน้าสำหรับดำเนินการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี 69

6) หน้าจอการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีสำเร็จแล้ว 70

4.2.2 แผนภาพแพ็คเกจและแผนภาพคาส (Package diagram and Class diagram) 70

1) คลาส FormImport 72

2) คลาส FormTranslation 72

3) คลาส FormShowOutputFIle 73

4) คลาส InputValidation 73

5) คลาส TPNModel 74

6) คลาส translatorTPN2EB 74

7) คลาส generateContext 75

8) คลาส tranlatorTPN2EB 75

9) คลาส MetaDataConstant 76

4.3 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ 76

4.3.1 ฮาร์ดแวร์ 76

4.3.2 ซอฟต์แวร์ 76

บทที่ 5 การทดสอบเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติ 77

5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ 77

5.2 แนวทางการทดสอบ 77

5.3 การทดสอบและประเมินผลระบบ 77

5.3.1 ประเมินผลการทำงานของเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติ โดยใช้กรณีทดสอบ สำหรับตรวจสอบไวยากรณ์และความหมายของข้อมูลนำเข้า 78

5.3.2 ประเมินผลเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติ โดยใช้กรณีศึกษาสำหรับตรวจสอบคุณสมบัติของผลลัพธ์การแปลงอีเวนท์บีโดยใช้เครื่องมือโรดิน 83

5.3.2.1 ทวนสอบการทำ งานของระบบไฟจราจร 83

1) ตรวจสอบคุณสมบัติด้านความปลอดภัยของระบบ (Safety Property)..... 85

2) ตรวจสอบหาภาวะติดตายของระบบ (Deadlock)..... 86

3) ตรวจสอบลำดับการยิงสำหรับแต่ละทรานซิชันในอีเวนท์บีต้องสอดคล้องกับนิยามของไทมด์เพทริเน็ต..... 86

5.3.2.2 ทวนสอบการทำงานของระบบการทำงานแบบพร้อมกัน (Concurrent system) 87

1) ตรวจสอบคุณสมบัติด้านความปลอดภัยของระบบ (Safety Property)..... 91

1) ตรวจสอบหาภาวะติดตายของระบบ (Deadlock)..... 91

2) ตรวจสอบลำดับการยิงสำหรับแต่ละทรานซิชันในอีเวนท์บีต้องสอดคล้องกับนิยามของไทมด์เพทริเน็ต..... 92

5.3.2.3 ทวนสอบการทำงานของระบบห่วงโซ่อุปทาน (Supply Chain)..... 94

1) รูปแบบเหตุและผล (Cause result pattern)..... 94

2) รูปแบบสาเหตุที่วนกลับ (Repeat cause one effect pattern)..... 94

3) รูปแบบการเกิดสาเหตุของหลายๆเหตุการณ์ เป็นสาเหตุทำให้เกิดเหตุการณ์หนึ่งเหตุการณ์ (N causes to one result pattern)..... 95

4) รูปแบบของหนึ่งเหตุการณ์เป็นสาเหตุทำให้เกิดหลายเหตุการณ์ (One cause to N results pattern)..... 96

5) ตรวจสอบคุณสมบัติด้านความปลอดภัยของระบบ (Safety Property)..... 101

6) ตรวจสอบหาภาวะติดตายของระบบ (Deadlock)..... 102

7) ตรวจสอบลำดับการยิงสำหรับแต่ละทรานซิชันในอีเวนท์บีต้องสอดคล้องกับนิยามของไทมด์เพทริเน็ต 103

บทที่ 6 สรุปผลการดำเนินงานวิจัย 105

6.1 สรุปผลงานวิจัย 105

6.2 ข้อจำกัดของเครื่องมือการแปลง 106

6.3 ข้อเสนอแนะ 106

รายการอ้างอิง 107

ภาคผนวก 110

1) ผลลัพธ์การแปลงอีเวนท์บีโค้ดที่สมบรูณ์ของแบบจำลองไทมด์เพทริเน็ตสำหรับระบบไฟจราจร 110

2) ผลลัพธ์การแปลงอีเวนท์บีโค้ดที่สมบรูณ์ของแบบจำลองไทมด์เพทริเน็ตสำหรับระบบการทำงานแบบพร้อมกัน 112

3) ผลลัพธ์การแปลงอีเวนท์บีโค้ดที่สมบรูณ์ของแบบจำลองไทมด์เพทริเน็ตสำหรับระบบห่วงโซ่อุปทาน 115

ประวัติผู้เขียนวิทยานิพนธ์ 118

สารบัญตาราง

ตารางที่ 2.1 องค์ประกอบเชิงสัญลักษณ์ของเพทริเน็ต [Choi, #7] 16
ตารางที่ 2.2 องค์ประกอบเชิงสัญลักษณ์กราฟิกของเพทริเน็ต [Choi, #7] 16
ตารางที่ 2.3 องค์ประกอบเชิงสัญลักษณ์ของไทมด์เพททริเน็ต [9] 19
ตารางที่ 2.4 องค์ประกอบของบริบท [Abrial, #15] 21
ตารางที่ 2.5 องค์ประกอบของแมชชีน [Abrial, #15] 22
ตารางที่ 2.6 รูปแบบทั่วไปของอีเวนท์ [Abrial, #15] 22
ตารางที่ 2.7 ตัวอย่างสัญกรณ์ทางคณิตศาสตร์ของอีเวนท์บี [Abrial, #15] 24
ตารางที่ 3.1 การอธิบายแผนภาพการนิยามเค้าร่างไทมด์เพทริเน็ตที่อยู่ในรูปแบบแฟ้มเอกสารเอกซ์เอ็มแอล 37
ตารางที่ 3.2 ความสัมพันธ์ของส่วนประกอบไทมด์เพทริเน็ตและส่วนประกอบเอกซ์เอ็มแอล 38
ตารางที่ 3.3 ความสัมพันธ์ไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎการแปลงส่วนประกอบเพลส 39
ตารางที่ 3.4 ความสัมพันธ์ไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎการแปลงส่วนประกอบทรานซิชัน 41
ตารางที่ 3.5 ความสัมพันธ์ระหว่างไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎการแปลงส่วนประกอบของการไหลของเส้นทางการย้ายสถานะ 41
ตารางที่ 3.6 ความสัมพันธ์ไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎการแปลงช่วงเวลาของแต่ละทรานซิชัน 42
ตารางที่ 3.7 ความสัมพันธ์ไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎกานิยามตัวแปรและตัวยืนยงในอีเวนท์บี 43
ตารางที่ 3.8 ความสัมพันธ์ไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎการแปลงเหตุการณ์เริ่มต้น 44
ตารางที่ 3.9 อธิบายสัญลักษณ์สำหรับอักขระที่ไม่ใช่แอสกี้กับค่าของชุดอักขระยูนิโค้ด 51
ตารางที่ 3.10 อธิบายสัญลักษณ์สำหรับแคลคูลัสบริสุทธิ์กับค่าของชุดอักขระยูนิโค้ด 51
ตารางที่ 3.11 อธิบายสัญลักษณ์สำหรับความสัมพันธ์และฟังก์ชันกับค่าของชุดอักขระยูนิโค้ด 52

ตารางที่ 3.12 อธิบายสัญลักษณ์สำหรับการประโยคเงื่อนไขกับค่าของชุดอักขระยูนิโค้ด 52

ตารางที่ 4.1 แสดงยูสเคสไอดีของยูสเคสทั้งหมด 57

ตารางที่ 4.2 รายละเอียดยูสเคสการนำเข้าแบบจำลองไทมด์เพทริเน็ตในรูปแบบแฟ้มเอกสาร เอกซ์เอ็มแอล 57

ตารางที่ 4.3 รายละเอียดยูสเคสการตรวจสอบความถูกต้องของรูปแบบการนำเข้าแบบจำลอง ไทมด์เพทริเน็ต 58

ตารางที่ 4.4 รายละเอียดยูสเคสการแปลงแบจำลองไทมด์เพทริเน็ตไปเป็นอีเวนท์บี 59

ตารางที่ 4.5 รายละเอียดยูสเคสการสกัดและเก็บข้องมูลส่วนประกอบไทมด์เพทริเน็ต 60

ตารางที่ 4.6 รายละเอียดยูสเคสการสร้างอีเวนท์บีโค้ด 61

ตารางที่ 4.7 รายละเอียดยูสเคสการเขียนผลลัพธ์การแปลงอีเวนท์บีให้อยู่ในแฟ้มเอกสารข้อความ . 62

ตารางที่ 4.8 รายละเอียดยูสเคสเพื่อแสดงหน้าต่างที่อยู่ของผลลัพธ์การแปลง 63

ตารางที่ 4.9 รายละเอียดยูสเคสการแสดงผลลัพธ์การแปลงอีเวนท์บี 64

ตารางที่ 5.1 เป้าหมายการทวนเพื่อตรวจสอบลำกับการยิงของทรานซิชันของระบบไฟจราจร 87

ตารางที่ 5.2 เป้าหมายการทวนเพื่อตรวจสอบลำกับการยิงของทรานซิชันของระบบการทำงาน แบบทำงานพร้อมกัน 93

ตารางที่ 5.3 เป้าหมายการทวนเพื่อตรวจสอบลำกับการยิงของทรานซิชันของระบบห่วงโซ่ อุปทาน 103

สารบัญรูป

รูปที่ 2.1 ตัวอย่างแบบจำลองเพทริเน็ต [Choi, #7] 17
รูปที่ 2.2 รูปแบบของเพทริเน็ต (Pattern of Petri Nets) [Choi, #7] 18
รูปที่ 2.3 ตัวอย่างแบบจำลองไทมด์เพทริเน็ต [9] 19
รูปที่ 2.4 ความสัมพันธ์ระหว่างบริบทและแมชชีน [Abrial, #15] 23
รูปที่ 2.5 ตัวอย่างเครื่องมือโรดิน 25
รูปที่ 2.6 ผลการตรวจสอบแบบสถิต 26
รูปที่ 2.7 ผลการสร้างข้อผูกพัน 26
รูปที่ 2.8 ผลการพิสูจน์ 27
รูปที่ 2.9 ส่วนต่อขยายทฤษฎีแซดทิสอะบิลิตี้โมดูโล 27
รูปที่ 2.10 ตัวอย่างการพิสูจน์ข้อผูกพันโดยเครื่องมือโรดิน 28
รูปที่ 2.11 ตัวอย่างการพิสูจน์ข้อผูกพันโดยส่วนต่อขยายเอสเอ็มที 28
รูปที่ 2.12 ตัวอย่างการใช้เครื่องมือโปรบีสำหรับการตรวจสอบภาวะติดตาย 29
รูปที่ 2.13 ตรรกศาสตร์เชิงเวลา 29
รูปที่ 3.1 แผนภาพกิจกรรมของวิธีดำเนินการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี 34
รูปที่ 3.2 ตัวอย่างแบบจำลองไทมด์เพทริเน็ต 35
รูปที่ 3.3 รูปแบบการเขียนป้ายระบุเปิดและป้ายระบุปิด 36
รูปที่ 3.4 รากของส่วนย่อย TPN ในการเขียนแบบจำลองไทมด์เพทริเน็ตในรูปแบบแฟ้มเอกซ์เอ็มแอล 36
รูปที่ 3.5 แผนภาพออกแบบเค้าร่างไทมด์เพทริเน็ตในรูปแบบแฟ้มเอกซ์เอ็มแอล 36
รูปที่ 3.6 ตัวอย่างรูปแบบแฟ้มเอกซ์เอ็มแอลขององค์ประกอบไทมด์เพทริเน็ต 38
รูปที่ 3.7 แผนภาพการทำงานของเหตุการณ์แบบไดนามิกภายในอีเวนท์บีโค้ด 45
รูปที่ 3.8 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ Enabled สำหรับอีเวนท์บีโค้ด 45

รูปที่ 3.9 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ Counting สำหรับอีเวนท์บีโค้ด 46

รูปที่ 3.10 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ Fire สำหรับอีเวนท์บีโค้ด ... 47

รูปที่ 3.11 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ resetIndividualTransition สำหรับอีเวนท์บีโค้ด 47

รูปที่ 3.12 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ resetConcurrentTransition สำหรับอีเวนท์บีโค้ด 48

รูปที่ 3.13 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ removeToken สำหรับอีเวนท์บีโค้ด 49

รูปที่ 3.14 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ e_DynamicTransition สำหรับ 50

รูปที่ 3.15 ตัวอย่างกฎการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี 50

รูปที่ 3.16 ตัวอย่างการแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบบริบทโดยใช้กฎการแปลงข้อที่ 1 ถึง กฎการแปลงข้อที่ 4 53

รูปที่ 3.17 ตัวอย่างการแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบแมชชีนโดยใช้กฎการแปลงข้อที่ 5 ถึง กฎการแปลงข้อที่ 6 53

รูปที่ 3.18 ตัวอย่างการแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบแมชชีนโดยใช้กฎการแปลงข้อที่ 7 สำหรับเหตุการณ์ Enable และ เหตุการณ์ Fired 54

รูปที่ 3.19 ตัวอย่างการแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบแมชชีนโดยใช้กฎการแปลงข้อที่ 7 สำหรับเหตุการณ์ e_DynamicTransition และ เหตุการณ์ Counting 54

รูปที่ 3.20 ตัวอย่างการแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบแมชชีนโดยใช้กฎการแปลงข้อที่ 7 สำหรับเหตุการณ์ resetIndividualTransition, เหตุการณ์ removeToken และ เหตุการณ์ resetConcurrentTransition 55

รูปที่ 4.1 แผนภาพยูสเคสแสดงขอบเขตการทำงานของระบบ 56

รูปที่ 4.2 แนวคิดการทำงานของเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บี 65

รูปที่ 4.3 แผนภาพกิจกรรมของเครื่องมือการแปลงแบบจำลองไทมด์เพทริเน็ตไปเป็นอีเวนท์บี 66

รูปที่ 4.4 หน้าจอนำเข้าจำลองไทมด์เพทริเน็ตที่อยู่ในรูปแบบของแฟ้มเอกสารเอกซ์เอ็มแอล..... 67
รูปที่ 4.5 หน้าจอสำหรับการหาที่อยู่ของแฟ้มเอกสารนำเข้าเอกซ์เอ็มแอลของส่วนประกอบไทมด์เพทริเน็ต..... 67
รูปที่ 4.6 หน้าจอแสดงส่วนประกอบที่ได้จากแฟ้มเอกสารนำเข้าเอกซ์เอ็มแอล..... 68
รูปที่ 4.7 หน้าจอการระบุตำแหน่งสำรับการบันทึกแฟ้มเอกสารข้อความของผลลัพธ์การแปลง 69
รูปที่ 4.8 หน้าสำหรับดำเนินการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี..... 69
รูปที่ 4.9 หน้าจอการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีสำเร็จแล้ว..... 70
รูปที่ 4.10 หน้าต่างที่อยู่ของผลลัพธ์การแปลง..... 70
รูปที่ 4.11 แผนภาพคลาสของการพัฒนาเครื่องมือการแปลง..... 71
รูปที่ 4.12 แผนภาพคลาสของคลาสของการนิยามค่าคงที่..... 71
รูปที่ 4.13 รายละเอียดคลาส FormImport..... 72
รูปที่ 4.14 รายละเอียดคลาส FormTranslation..... 72
รูปที่ 4.15 รายละเอียดคลาส FormShowOutputFIle..... 73
รูปที่ 4.16 รายละเอียดคลาส InputValidation..... 73
รูปที่ 4.17 รายละเอียดคลาส TPNModel..... 74
รูปที่ 4.18 รายละเอียดคลาส translatorTPN2EB..... 74
รูปที่ 4.19 รายละเอียดคลาส generateContext..... 75
รูปที่ 4.20 รายละเอียดคลาส tranlatorTPN2EB..... 75
รูปที่ 4.21 รายละเอียดคลาส MetaDataConstant..... 76
รูปที่ 5.1 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีที่นำเข้าผิดเอกสารที่ไม่ใช่แบบจำลองไทมด์เพทริเน็ต..... 79
รูปที่ 5.2 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีที่นำเข้าผิดเอกสารที่ไม่ใช่แบบจำลองไทมด์เพทริเน็ต..... 79
รูปที่ 5.3 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีโครงสร้างเอกซ์เอ็มแอลผิด..... 80

รูปที่ 5.4 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีโครงสร้างเอกซ์เอ็มแอลผิด 80

รูปที่ 5.5 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้เปลี่ยนแปลงโครงสร้างของเอกซ์เอ็มแอล 80

รูปที่ 5.6 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้เปลี่ยนแปลงโครงสร้างของเอกซ์เอ็มแอล 81

รูปที่ 5.7 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำ 81

รูปที่ 5.8 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำ 81

รูปที่ 5.9 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำกับชื่อทรานซิชัน 81

รูปที่ 5.10 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำกับชื่อทรานซิชัน 82

รูปที่ 5.11 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุเส้นทางจากทรานซิชันไปทรานซิชัน 82

รูปที่ 5.12 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุเส้นทางจากทรานซิชันไปทรานซิชัน 82

รูปที่ 5.13 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุไม่ระบุเส้นทาง 82

รูปที่ 5.14 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุไม่ระบุเส้นทาง 83

รูปที่ 5.15 ระบบไฟจราจรที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ต 83

รูปที่ 5.16 แฟ้มเอกสารเอกซ์เอ็มแอลที่ระบุส่วนประกอบของระบบไฟจราจรที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ต 84

รูปที่ 5.17 ผลลัพธ์การแปลงส่วนประกอบบริบทของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบไฟจราจร 84

รูปที่ 5.18 ผลลัพธ์การแปลงส่วนประกอบแมชชีนของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบไฟจราจร 85

รูปที่ 5.19 ผลลัพธ์ตรวจสอบหาภาวะติดตายของระบบไฟจราจร 86

รูปที่ 5.20 ผลลัพธ์การตรวจสอบเวลาการยิงสำหรับแต่ละทรานซิชัน 87

รูปที่ 5.21 ระบบการทำงานแบบทำงานพร้อมกันที่อยู่ในรูปแบบของจำลองไทมด์เพทริเน็ต..... 88

รูปที่ 5.22 แฟ้มเอกสารเอกซ์เอ็มแอลที่ระบุส่วนประกอบของระบบการทำงานแบบพร้อมกันที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ต..... 89

รูปที่ 5.23 ผลลัพธ์การแปลงส่วนประกอบบริบทของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบการทำงานแบบพร้อมกัน..... 90

รูปที่ 5.24 ผลลัพธ์การแปลงส่วนประกอบแมชชีนของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบการทำงานแบบพร้อมกัน..... 90

รูปที่ 5.25 ผลลัพธ์ตรวจสอบหาภาวะติดตายของระบบการทำงานแบบทำงานพร้อมกัน..... 92

รูปที่ 5.26 ผลลัพธ์การตรวจสอบเวลาการยิงสำหรับแต่ละทรานซิชันของระบบแบบการทำงานพร้อมกัน..... 93

รูปที่ 5.27 ตัวอย่างของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบของเหตุผล..... 94

รูปที่ 5.28 ผลลัพธ์อีเวนท์บีโค้ดของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบของเหตุผล..... 94

รูปที่ 5.29 ตัวอย่างของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบสาเหตุที่เกิดผลซ้ำ..... 95

รูปที่ 5.30 ผลลัพธ์อีเวนท์บีโค้ดของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบสาเหตุที่เกิดผลซ้ำ..... 95

รูปที่ 5.31 ตัวอย่างของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบการเกิดสาเหตุของหลายๆเหตุการณ์ เป็นสาเหตุทำให้เกิดเหตุการณ์หนึ่งเหตุการณ์..... 95

รูปที่ 5.32 ผลลัพธ์อีเวนท์บีโค้ดของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบการเกิดสาเหตุของหลายๆเหตุการณ์ เป็นสาเหตุทำให้เกิดเหตุการณ์หนึ่งเหตุการณ์..... 96

รูปที่ 5.33 ตัวอย่างของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบของหนึ่งเหตุการณ์เป็นสาเหตุทำให้เกิดหลายเหตุการณ์..... 96

รูปที่ 5.34 ผลลัพธ์อีเวนท์บีโค้ดของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบของหนึ่งเหตุการณ์เป็นสาเหตุทำให้เกิดหลายเหตุการณ์..... 97

รูปที่ 5.35 ระบบห่วงโซ่อุปทานที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ต..... 98

รูปที่ 5.36 แฟ้มเอกสารเอกซ์เอ็มแอลที่ระบุส่วนประกอบของระบบการทำงานแบบพร้อมกันที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ต..... 99

รูปที่ 5.37 ผลลัพธ์การแปลงส่วนประกอบบริบทของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบห่วงโซ่อุปทาน 100

รูปที่ 5.38 ผลลัพธ์การแปลงส่วนประกอบแมชชีนของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบห่วงโซ่อุปทาน..... 101

รูปที่ 5.39 ผลลัพธ์ตรวจสอบหาภาวะติดตายของระบบห่วงโซ่อุปทาน 103

รูปที่ 5.40 ผลลัพธ์การตรวจสอบเวลาการยิงสำหรับแต่ละทรานซิชันของระบบห่วงโซ่อุปาทาน 104

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ระบบซอฟต์แวร์ในปัจจุบันได้พัฒนาเพื่อเป็นตัวช่วยให้มนุษย์มีความสะดวกสบายมากยิ่งขึ้น ส่งผลให้อุตสาหกรรมซอฟต์แวร์มีความต้องการพัฒนาระบบที่จะตอบสนองความต้องการของผู้รับบริการหรือลูกค้า ระบบที่ถูกพัฒนาออกมาในท้องตลาดมีหลายหลากรูปแบบ ซึ่งความซับซ้อนของระบบเกิดขึ้นได้กับระบบที่มีขนาดเล็กและขนาดใหญ่แต่ระบบที่มีขนาดใหญ่มักจะมีความซับซ้อนสูงและมีโอกาสที่จะเกิดข้อผิดพลาดได้ง่าย ผลกระทบของข้อผิดพลาดที่เกิดขึ้นหลังจากผ่านกระบวนการพัฒนาซอฟต์แวร์จะมีอัตราการสูญเสียมากกว่าข้อผิดพลาดที่เกิดขึ้นในก่อนกระบวนการพัฒนาไม่ว่าจะเป็น เวลา (Time) ค่าใช้จ่าย (Cost) และทรัพยากร (Resource) ที่สูญเสียไป ดังนั้นอุตสาหกรรมซอฟต์แวร์จึงตระหนักได้ว่าการตรวจสอบการออกแบบซอฟต์แวร์ก่อนอาจส่งผลต่อการลดเวลาในการพัฒนาและค่าใช้จ่าย โดยเฉพาะอย่างยิ่งสำหรับระบบซอฟต์แวร์แบบเรียลไทม์ (Real time system) ควรได้รับการป้องกันความปลอดภัยและตรวจสอบความถูกต้องของการออกแบบซอฟต์แวร์เป็นไปตามข้อกำหนดความต้องการหรือไม่ การทวนสอบเชิงรูปนัย (Formal Verification) ของระบบซอฟต์แวร์แบบเรียลไทม์จึงเป็นทางเลือกที่ได้รับความนิยมและสามารถตรวจสอบพฤติกรรมที่มีเวลาเข้ามาเกี่ยวข้องได้อย่างละเอียด ในขณะที่การทดสอบซอฟต์แวร์แบบเรียลไทม์ดำเนินการเพียงขั้นตอนในการออกแบบเท่านั้น การทวนสอบเชิงรูปนัยเป็นกระบวนการพิสูจน์เพื่อให้แน่ใจว่าระบบซอฟต์แวร์ที่เป็นเป้าหมายมีคุณสมบัติปลอดภัยโดยใช้วิธีทางคณิตศาสตร์ การทวนสอบส่วนใหญ่เริ่มต้นด้วยการสร้างตัวแทนนามธรรมเพื่อจำลองระบบโลกแห่งความจริงเรียกว่าวิธีเชิงรูปนัย (Formal method) แบบจำลองที่เป็นไปได้มีหลากหลายรูปแบบที่ได้พิจารณากระบวนการทางซอฟต์แวร์โดยบูรณการวิธีเชิงรูปนัยเข้าด้วยกัน เช่น การแปลงภาษาแซดไปเป็นเพทริเน็ตเพื่อสร้างแบบจำลองของระบบซอฟต์แวร์ที่มีความปลอดภัย [1] การแปลงจากเพทริเน็ตไปเป็นอีเวนท์บี โดยใช้เทคนิคการฝังความหมายโดยใช้นามธรรมบีเป็นตัวกลางสำหรับการแปลงจากเพทริเน็ตไปเป็นอีเวนท์บีและแนวคิดการแปลงจากอีเวนท์บีไปยังเพทริเน็ตสำหรับการประเมินเชิงคุณภาพของระบบซอฟต์แวร์ที่เป็นเป้าหมาย อย่างไรก็ตามกระบวนการเชิงรูปนัยจำเป็นต้องมีหลักฐานยืนยันว่าได้รับการตรวจสอบความปลอดภัยของข้อกำหนดความต้องการเรียบร้อยแล้ว [2] ดังนั้นงานวิจัยนี้จึงได้นำเสนอกฎการแปลงโดยใช้สัญกรณ์พื้นฐานของไทมด์เพทริเน็ตซึ่งได้รับการพิจารณาและแปลงไปในส่วนของโปรแกรมในอีเวนท์บี

แบบจำลองทั่วไปของเพทริเน็ต [3] เป็นวิธีการเชิงรูปนัยที่จำลองพฤติกรรมของระบบที่ทำงานพร้อมกันทำให้สามารถวิเคราะห์พฤติกรรมเชิงระบบและเชิงปริมาณได้ เครื่องมือจำลองแบบเพทริเน็ต [4] ถูกเสนอและพัฒนาขึ้นเพื่อตรวจสอบความถูกต้องของเพทริเน็ต เมื่อต้องการจัดการกับระบบที่มีระยะเวลาเข้ามาเกี่ยวข้องของไทมด์เพทริเน็ต [5, 6] จึงถูกนำมาใช้แทนเพทริเน็ตแบบดั้งเดิม ไทมด์เพทริเน็ตใช้สัญลักษณ์กราฟิกสำหรับจำลองโครงสร้างระบบทำให้สามารถเข้าใจพฤติกรรมของระบบเป้าหมายได้ง่าย แต่ยังขาด

การแสดงในส่วนของข้อมูลที่ใช้ภายในระบบซอฟต์แวร์และไม่ได้รับการปรับแต่งใดๆ ในกระบวนการพัฒนาซอฟต์แวร์ การจำลองพฤติกรรมของเพทริเน็ตยังต้องใช้เวลานานสำหรับการตรวจสอบอย่างละเอียดและไทมด์เพทริเน็ต หากระบบมีขนาดใหญ่และมีความซับซ้อนมากอาจจะเกิดปัญหาการระเบิดของสถานะ (State explosion) ทำให้ไม่สามารถทวนสอบระบบได้สะดวกและยังไม่ได้รับการตรวจสอบทฤษฎีความแซดทิสอะบิลิตี้โมดูโล (Satisfiability Modulo Theory: SMT) อีกหนึ่งทางเลือกของวิธีเชิงรูปนัยพบว่าอีเวนท์บี [7] ซึ่งถูกพัฒนามาจากบีเมทอดเป็นวิธีเชิงรูปนัยที่นิยมสำหรับการสร้างแบบจำลองและการวิเคราะห์ระดับระบบซึ่งสนับสนุนการใช้การปรับแต่งของถึงระบบซอฟต์แวร์แบบเรียลไทม์และใช้เทคนิคการพิสูจน์โดยทฤษฎีทางคณิตศาสตร์ (Theorem proving) ซึ่งจะทำให้ไม่เกิดปัญหาการระเบิดของสถานะในระหว่างการทวนสอบ นอกจากนี้อีเวนท์บียังมีเครื่องมือสนับสนุนการตรวจสอบระบบที่ชื่อว่าเครื่องมือโรดินซึ่งมีการพิสูจน์ทางคณิตศาสตร์และมีการตรวจสอบทฤษฎีความพึงพอใจที่ได้รับความนิยมสำหรับการทวนสอบเชิงรูปนัยซึ่งเป็นการเพิ่มประสิทธิภาพของการพิสูจน์ได้อย่างอัตโนมัติ งานวิจัยนี้สนใจการทวนสอบระบบโดยใช้เทคนิคการพิสูจน์ทางคณิตศาสตร์เพื่อไม่ให้เกิดปัญหาการระเบิดของสถานะโดยอีเวนท์บีแต่การเขียนแบบจำลองในโปรแกรมของอีเวนท์เป็นเรื่องยากจำเป็นต้องใช้พื้นฐานทางคณิตศาสตร์เพื่อทำความเข้าใจการทำงานของระบบและมีข้อจำกัดของรูปแบบที่เกี่ยวกับเวลา [8] อย่างไรก็ตามคุณสมบัติของไทมด์เพทริเน็ตที่มีการแสดงแบบจำลองเชิงกราฟิกทำให้มีความสามารถทำความเข้าใจ (Understandability) การทำงานของระบบได้ง่าย

ดังนั้นงานวิจัยนี้จึงใช้แบบจำลองไทมด์เพทริเน็ตที่เป็นต้นแบบจำลองของการทำงานระบบแปลงให้เป็นอีเวนท์บีเพื่อตรวจสอบระบบได้อย่างละเอียดโดยใช้เทคนิคการพิสูจน์และตรวจสอบทฤษฎีความพึงพอใจของระบบซึ่งงานวิจัยนี้จะสนใจค่าน้ำหนักของการเปลี่ยนสถานะมีค่าเท่ากับหนึ่งเท่านั้นและนำเสนอกฎการแปลงโดยใช้สัญกรณ์พื้นฐานของไทมด์เพทริเน็ตแปลงเป็นโปรแกรมของอีเวนท์บี โดยรายละเอียดของผลลัพธ์การแปลงจะถูกรวมไว้ในส่วนประกอบบริบทและส่วนประกอบแมชชีน

1.2 วัตถุประสงค์

1. เพื่อนำเสนอกฎการแปลงไทมด์เพทริเน็ตเป็นอีเวนท์บี
2. เพื่อพัฒนาเครื่องมือสนับสนุนแปลงไทมด์เพทริเน็ตเป็นอีเวนท์บี

1.3 ขอบเขตการดำเนินงาน

1. ไทมด์เพทริเน็ตที่นำมาแปลงมีลักษณะดังต่อไปนี้
   1.1. น้ำหนักโทเค็นสำหรับการย้ายโทเค็นของอินพุตเพลสมีค่าเท่ากับหนึ่งเท่านั้น
   1.2. น้ำหนักโทเค็นสำหรับการย้ายโทเค็นของเอาต์พุตเพลสมีค่าเท่ากับหนึ่งเท่านั้น
   1.3. ไทมด์เพทริเน็ตไม่เรียกใช้ติดต่อกับไทมด์เพทริเน็ตภายนอกสามารถจบได้ด้วยตัวเอง
   1.4. องค์ประกอบของไทมด์เพทริเน็ตมีจำนวนห้าองค์ประกอบ ซึ่งจะต้องประกอบด้วย เพลส, ทรานซิชัน, เส้นทาง, น้ำหนักของเส้นทางและมาร์กกิง

2. อีเวนท์บีที่แปลงได้มีลักษณะดังต่อไปนี้
   2.1. อีเวนท์บีมีหนึ่งบริบทและมีหนึ่งแมชชีนที่สามารถมองเห็นบริบทที่ถูกแปลงออกมาได้
   2.2. แมชชีนในอีเวนท์บีไม่ติดต่อกับแมชชีนและบริบทภายนอก
   2.3. อีเวนท์บีมีการพิสูจน์ความถูกต้องจากโปรแกรมโรดิน
3. กฎการแปลงไทมด์เพทริเน็ตเป็นอีเวนท์บี
   3.1. กฎการแปลงที่นำเสนออยู่ในรูปแบบของตารางความสัมพันธ์ระหว่างไทมด์เพทริเน็ตกับอีเวนท์บี
   3.2. สามารถแปลงได้ครบทุกห้าองค์ประกอบของไทมด์เพทริเน็ต
4. เครื่องมือสนับสนุนการแปลงมีขีดความสามารถดังต่อไปนี้
   4.1. ข้อมูลนำเข้าเป็นไทมด์เพทริเน็ตที่อยู่ในรูปแบบของแฟ้มเอกซ์เอ็มแอล
   4.2. สามารถแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บี โดยประกอบด้วยสองส่วนประกอบคือ หนึ่งบริบทและหนึ่งแมชชีน
   4.3. ผลลัพธ์การแปลงได้รับการตรวจสอบความถูกต้องโดยเครื่องมือโรดิน
   4.4. สามารถตรวจสอบไวยากรณ์และความหมายของแบบจำลองไทมด์เพทริเน็ตก่อนเริ่มต้นแปลง หากพบข้อผิดพลาดมีข้อความแจ้งเตือนและจบการทำงาน
5. ใช้กรณีศึกษาเพื่อประเมินผลการทำงานของเครื่องมืออย่างน้อยสามกรณีศึกษา
6. กรณีทดสอบเพื่อประเมินผลการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี
   6.1. ตรวจสอบไวยากรณ์และความหมายของแบบจำลองไทมด์เพทริเน็ต
   6.2. ตรวจสอบความครบถ้วนขององค์ประกอบของไทมด์เพทริเน็ตที่อยู่ในผลลัพธ์การแปลงอีเวนท์บี
   6.3. ตรวจสอบคุณสมบัติด้านความปลอดภัยของระบบ (Safety Property)
   6.4. ตรวจสอบหาภาวะติดตายของระบบ (Deadlock)
   6.5. ตรวจสอบเวลาการยิงสำหรับแต่ละทรานซิชันในอีเวนท์บีต้องสอดคล้องกับนิยามของไทมด์เพทริเน็ต (Firing sequence of each transition)

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษารูปแบบไวยากรณ์ของอีเวนท์บี
2. ทำความเข้าใจเกี่ยวกับข้อจำกัดของภาษาอีเวนท์บี
3. ศึกษาสัญลักษณ์ของไทมด์เพทริเน็ต
4. ออกแบบกฎการแปลงอีเวนท์บีสำหรับการแปลงแบบจำลองไทมด์เพทริเน็ต
5. พัฒนาเครื่องมืออัตโนมัติสำหรับการแปลงไทมด์เพทริเน็ตไปเป็นภาษาอีเวนท์บี
6. ศึกษาวิธีการใช้งานเครื่องมือโรดิน
7. ประเมินความถูกต้องของผลลัพธ์การแปลง
8. สรุปผลวิจัยและข้อเสนอแนะ
9. จัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้กฎการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี
2. ได้เครื่องมือสนับสนุนแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีโดยอัตโนมัติ

1.6 บทความวิจัยที่ได้รับการตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นวารสารวิชาการและบทความวิชาการ มีดังต่อไปนี้

1. วารสารวิชาการเรื่อง “Automatic Transformation of Ordinary Timed Petri Nets into Event-B for Formal Verification” ฉบับเดือนกรกฎาคม ปี 2561 (vol. 22, no.4, ISSN: 0125-8281, DOI: 10.4186) โดย Engineering Journal คณะวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัย ณ จังหวัดกรุงเทพมหานคร ประเทศไทย

2. บทความวิชาการเรื่อง “Event-B Formalization of Basic Supply Chain Patterns” ในงานประชุมวิชาการ 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (IEEE/ACIS SNPD 2018) จัดโดย IEEE and ACIS, USA เมื่อวันที่ 27-29 มิถุนายน 2561 ณ ปูซาน ประเทศเกาหลีใต้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

ผู้วิจัยได้ศึกษาค้นคว้าเกี่ยวกับทฤษฎีที่เกี่ยวข้องกับวิธีการทวนสอบที่ใช้ในงานวิจัยคือเพทริเน็ต (Petri Nets) ไทมด์เพทริเน็ต (Time in Petri Nets) และอีเวนท์บี (Event-B) ซึ่งได้อธิบายรายละเอียดในลำดับถัดไป

2.1.1 เพทริเน็ต (Petri Nets)

ในปี 1962 คาร์ลอดัมเพทริ (Carl Adam Petri) [3] ได้คิดค้นเพทริเน็ตโดยมีวัตถุประสงค์เพื่ออธิบายกระบวนการทางเคมี ซึ่งเพทริเน็ตเป็นวิธีการทวนสอบเชิงรูปนัยแบบโมเดลเช็คกิงที่สำหรับระบบที่ทำงานพร้อมกัน โดยการสร้างแบบจำลองสำหรับการทวนสอบแบบจำลองของระบบที่ทำงานพร้อมกันเพื่อจำลองพฤติกรรมการทำงานของระบบ โดยใช้สัญลักษณ์กราฟิกจำลองพฤติกรรมของของระบบได้หลากหลายสถานการณ์ เช่น วงจรดิจิตอลหรือระบบการผลิตหรือระบบปฏิบัติการ เป็นต้น การใช้สัญลักษณ์กราฟิกเป็นตัวแทนของระบบทำให้ผู้วิเคราะห์ระบบสามารถเข้าใจการทำงานของระบบได้ง่ายและยังสามารถตรวจสอบคุณสมบัติของระบบ เช่น ความขัดแย้งหรือภาวะติดตาย (Deadlocks) สัญลักษณ์กราฟิกเป็นตัวแทนของพฤติกรรมระบบโดยเพลส (Place) อธิบายเงื่อนไขของการเกิดเหตุการณ์หรือกิจกรรมและในส่วนของทรานซิชัน (Transition) อธิบายเหตุการณ์เมื่อเงื่อนไขของการเกิดเหตุการณ์ที่เป็นจริง โดยมีเส้นเชื่อม (Arcs) ระหว่างเพลสไปยังทรานซิชันหรือเรียกว่า อินพุตเพลส (Input Place) และจากทรานซิชันไปยังเพลสหรือเรียกว่า เอาต์พุตเพลส (Output Place) โดยหัวลูกศรเพื่ออธิบายลำดับการทำงานก่อนหลังได้ การกำหนดการเหตุการณ์ที่เกิดขึ้นจะแทนด้วยลูกวงกลมสีดำเรียกว่า โทเค็น (Token) โดยที่แต่ละโทเค็นจะถูกระบุให้อยู่ใน เพลส เพื่ออธิบายเหตุการณ์ที่พร้อมจะเกิดเหตุการณ์ การกำหนดเหตุการณ์เริ่มต้นจะมีการระบุระบุโทเค็นลงไปในเพลสเริ่มต้นของระบบเรียกว่า มาร์กกิง (Initial Marking) เมื่อเพลสมีโทเค็นครบตามจำนวนที่ทรานซิชันต้องการในแต่ละอินพุตเพลส แสดงว่าเงื่อนไขเป็นจริงจะเหตุการณ์ของทรานซิชันนั้นถูกเปิดใช้ (Enabled) เมื่อถูกเปิดใช้โทเค็นจะถูกลบออกจากอินพุตเพลสและโทเค็นจะถูกเพิ่มในเอาต์พุตเพลสเรียกว่า การยิง (Firing) องค์ประกอบของเพทริเน็ตที่เป็นรูปแบบดั้งเดิม (Timeless) มีทั้งหมดมีทั้งหมด 5 องค์ประกอบ (5-tuple) คือ $N = (P, T, F, V, m_0)$ อธิบายองค์ประกอบเชิงสัญลักษณ์ได้ใน

ตารางที่ 2.1

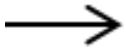
ตารางที่ 2.1 องค์ประกอบเชิงสัญลักษณ์ของเพทริเน็ต [3]

สัญลักษณ์	คำอธิบาย
$P = \{p1,...,pk\}$ โดยที่ $P \cap T = \emptyset$ และ $P \cup T \neq \emptyset$	เพลส (Place) คือ ชุดของจำนวนจำกัดของเงื่อนไขที่ใช้กำหนดการเกิดเหตุการณ์
$T = \{t1,..., tn\}$ โดยที่ $P \cap T = \emptyset$ และ $P \cup T \neq \emptyset$	ทรานซิชัน (Transition) คือ ชุดของจำนวนจำกัดของเหตุการณ์ที่จะเกิดขึ้น
$F \subseteq (P \times T) \cup (T \times P)$ pre-places $(t^-) := \{p \mid p \in P \wedge (p, t) \in F\}$ post-places $(t^+) := \{p \mid p \in P \wedge (t, p) \in F\}$	เส้นทาง (Flow relation หรือ Arcs) คือ ชุดของเส้นทางก่อนเกิดเหตุการณ์จากเพลสไปยังทรานซิชัน (Input Places) และชุดของเส้นทางหลังเกิดเหตุการณ์จากทรานซิชันไปยังเพลส เอาต์พุตเพลส (Output Places)
$V : F \longrightarrow N^+$	น้ำหนักของเส้นทาง (Weight of arcs)
$M_0 : P \longrightarrow N$ $m_0 = M_0(p_i)$	มาร์กกิง (Initial Marking) คือ การกำหนดจำนวนโทเค็นที่เพลสใดๆ ซึ่ง m_0 คือการระบุโทเค็นครั้งแรกเพื่อกำหนดการเกิดเหตุการณ์เริ่มต้น

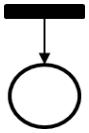
จากองค์ประกอบเชิงสัญลักษณ์ที่อธิบายในรูปแบบของคำบรรยายสามารถแปลงเป็นสัญลักษณ์กราฟิกได้ดังตารางที่ 2.2

ตารางที่ 2.2 องค์ประกอบเชิงสัญลักษณ์กราฟิกของเพทริเน็ต [3]

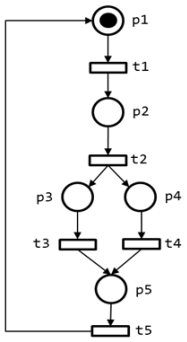
สัญลักษณ์กราฟิก	คำอธิบาย
	เพลส (Places) คือ ชุดของจำนวนจำกัดของเงื่อนไขที่ใช้กำหนดการเกิดเหตุการณ์
	ทรานซิชัน (Transition) คือ ชุดของจำนวนจำกัดของเหตุการณ์ที่จะเกิดขึ้น
	โทเค็น (Token) คือ ตัวกำหนดเหตุการณ์ที่จะเกิดขึ้นเมื่อเงื่อนไขเพลสเป็นจริง

สัญลักษณ์กราฟิก	คำอธิบาย
	เส้นทาง (Arc) คือ การไหลของเส้นทางการทำงานของระบบโดยมีหัวลูกศรเป็นตัวกำหนดทิศทาง

ตารางที่ 2.2 องค์ประกอบเชิงสัญลักษณ์กราฟิกของเพทริเน็ต [3] (ต่อ)

สัญลักษณ์กราฟิก	คำอธิบาย
	อินพุตเพลส (Input Places) คือ ชุดของเส้นทางก่อนเกิดเหตุการณ์จากเพลสไปยังทรานซิชัน
	เอาต์พุตเพลส (Output Places) คือ ชุดของเส้นทางหลังเกิดเหตุการณ์จากทรานซิชันไปยังเพลส
	มาร์กกิง (Initial Marking) คือ การระบุเหตุการณ์ที่จะเกิดขึ้น โดยการระบุโทเค็นที่อยู่ในเพลส

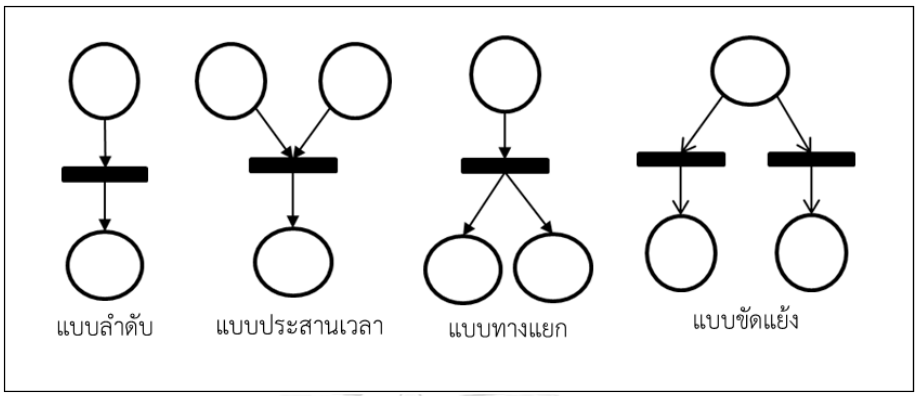
เพทริเน็ตสามารถเขียนออกมาให้อยู่ในรูปแบบเชิงสัญลักษณ์และเชิงกราฟิกได้ดังตัวอย่างแบบจำลองเพทริเน็ตใน รูปที่ 2.1

สัญลักษณ์กราฟิก	ข้อกำหนดโครงสร้างสัญลักษณ์
	TPNs=(P, T, F, V, m_0, D) P={p1, p2, p3, p4, p5} T={t1, t2, t3, t4, t5} F ={(p_1,t_1),(p_2,t_2),(p_3,t_3),(p_4,t_4),(p_5,t_5), (t_1,p_2),(t_2,p_3),(t_2,p_4),(t_3,p_5),(t_4,p_5),(t_5,p_1)} V = F {(p_1,t_1) →1,(p_2,t_2) →1,(p_3,t_3) →1, (p_4,t_4) →1, (p_5,t_5) →1,(t_1,p_2) →1, (t_2,p_3) →1,(t_2,p_4) →1, (t_3,p_5) →1, (t_4,p_5) →1,(t_5,p_1) →1} m_0=(1, 0, 0, 0, 0, 0)

รูปที่ 2.1 ตัวอย่างแบบจำลองเพทริเน็ต [3]

2.1.1.1 รูปแบบของเพทริเน็ต (Pattern of Petri Nets)

รูปแบบของเพทริเน็ตมี 4 รูปแบบ คือ แบบลำดับ (Sequence), แบบประสานเวลา (Join หรือ Synchronization), แบบทางแยก (Fork) และแบบขัดแย้ง (Decision หรือ Conflict) แสดงดังรูปที่ 2.2



รูปที่ 2.2 รูปแบบของเพทริเน็ต (Pattern of Petri Nets) [3]

2.1.1.2 พฤติกรรมของเพทริเน็ต (Behavior of Petri Nets)

1) เปิดใช้งานทรานซิชัน (Enabled Transitions)

พฤติกรรมของทรานซิชันถูกเปิดใช้งาน เมื่อเพลสมีจำนวนของโทเค็นเพียงพอตามค่าน้ำหนักที่อินพุตเพลสต้องการ

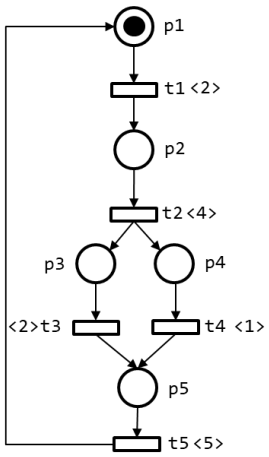
2) ยิงทรานซิชัน (Firing Transitions)

พฤติกรรมของทรานซิชันเมื่อถูกเปิดใช้งานแล้ว โทเค็นที่อยู่ในอินพุตเพลสจะถูกลบออกและโทเค็นจะถูกเพิ่มที่เอาต์พุตเพลสของทรานซิชันนั้น

2.1.1.3 ไทมด์เพทริเน็ต (Timed Petri Nets)

ระบบในโลกของความจริงการเกิดเหตุการณ์หรือกิจกรรมใดๆ จำเป็นต้องมีเรื่องเวลา (Times) และลำดับความสำคัญ (Priority) เพื่อให้มีความมั่นใจว่าทุกส่วนประกอบมีพฤติกรรมตรงตามข้อกำหนดของระบบทำงานแบบทันที (Real-time system) ดังนั้นจึงมีการนำเวลาเป็นตัวกำหนดช่วงเวลาของการเปลี่ยนสถานะของโทเค็น

ไทมด์เพทริเน็ตหรือชื่อย่อคือทีพีเอ็น (TPN) คิดค้นโดย Ramchandani [9] เป็นส่วนที่ขยายต่อเพทริเน็ตที่เป็นรูปแบบดั้งเดิม โดยการเพิ่มระยะเวลาให้สำหรับแต่ละทรานซิชันเพื่ออธิบายถึงระยะเวลาก่อนที่จะเกิดกระบวนการยิงโทเค็นเพื่อเปลี่ยนสถานะหนึ่งไปอีกสถานะหนึ่งโดยการย้ายโทเค็นจากอินพุตเพลสไปยังเอาต์พุตเพลส ระยะเวลาของการยิงแต่ละทรานซิชัน $d(t_i)$ ถูกเขียนขึ้นในแต่ละการเปลี่ยนแปลงภายในวงเล็บมุมแสดงตัวอย่างในรูปที่ 2.3



รูปที่ 2.3 ตัวอย่างแบบจำลองไทมด์เพทริเน็ต [9]

ไทมด์เพทริเน็ตรูปแบบดั้งเดิมที่มีรูปแบบที่มีการกำหนดระยะเวลาการยิงให้ทรานซิชัน มีทั้งหมดมีทั้งหมด 6 องค์ประกอบคือ TPN = (P, T, F, V, m_0, D) อธิบายองค์ประกอบเชิงสัญลักษณ์ได้ตารางที่ 2.3

ตารางที่ 2.3 องค์ประกอบเชิงสัญลักษณ์ของไทมด์เพททริเน็ต [9]

สัญลักษณ์	คำอธิบาย
(P, T, F, V, m0)	เพทริเน็ตรูปแบบดั้งเดิมจะสามารถแทนด้วย S(N)
$D : T \longrightarrow N^+$	ฟังก์ชันเวลาล่าช้าของเน็ต N สำหรับแต่ละทรานซิชันสามารถยิงโทเค็นได้ขึ้นอยู่กับจำนวนธรรมชาติ
$d_i := D(t_i)$	ระยะเวลาการยิงของทรานซิชัน t_i

2.1.1.4 พฤติกรรมของไทมด์เพทริเน็ต (Behavior of timed Petri net)

1) เปิดใช้งานทรานซิชัน (Enabled Transitions)

พฤติกรรมของทรานซิชันถูกเปิดใช้งานจะเหมือนกับเพทริเน็ตแบบดั้งเดิมคือเมื่อเพลสมีจำนวนของโทเค็นเพียงพอตามค่าน้ำหนักที่อินพุตเพลสต้องการ

2) ยิงทรานซิชัน (Firing Transitions)

พฤติกรรมการยิงทรานซิชันจะแตกต่างกับเพทริเน็ตแบบดั้งเดิม เมื่อทรานซิชันถูกเปิดใช้งานจะเกิดการนับเวลาตามระยะเวลาที่ถูกกำหนดไว้ของแต่ละทรานซิชันก่อนที่จะเกิดการยิง เมื่อตัวนับเวลาเท่ากับระยะเวลาที่กำหนดจะเกิดการยิงทรานซิชัน โดยการเคลื่อนโทเค็นที่อยู่ในอินพุตเพลสไปที่เอาต์พุตเพลส

2.1.2 อีเวนท์บี (Event-B)

ในปี 1988 Jean-Raymond Abrial ได้พัฒนาบีเมทอด (B-Method) [10] โดยมีจุดประสงค์เพื่อทวนสอบกระบวนการพัฒนาซอฟต์แวร์ (Development process) ของระบบแบบเป็นลำดับ (Sequential System) เป็นเรื่องยากที่จะทวนสอบระบบที่มีความซับซ้อนหรือระบบที่ทำงานพร้อมกัน ในปี 2010 อีเวนท์บี [7] ถูกพัฒนาจากผู้พัฒนาเดียวกันกับบีเมทอดซึ่งมีจุดประสงค์เพื่ออธิบายข้อกำหนดและสร้างแบบจำลองระดับระบบ (System Level) หนึ่งในความแตกต่างหลักระหว่างบีเมทอดและอีเวนท์บี คือ บีเมทอดจะเกี่ยวข้องกับตัวดำเนินการและอีเวนท์บีจะเกี่ยวข้องกับเหตุการณ์ แต่ละตัวดำเนินการในบีเมทอดจะถูกกำหนดพร้อมกับเงื่อนไขก่อน (Pre-Condition) และเมื่อเงื่อนไขเป็นจริง ตัวดำเนินการในบีเมทอดจะเกิดขึ้น แต่ในทางกลับกันแต่ละเหตุการณ์ในอีเวนท์บีจะถูกกำหนดพร้อมกับตัวป้องกัน (Guard) เมื่อตัวป้องกันเป็นจริงแต่ละเหตุการณ์ถึงจะเกิดขึ้นได้ และอีกหนึ่งสิ่งที่มีความแตกต่างกันคือ เงื่อนไขก่อนในบีเมทอดยังมีจุดอ่อนสำหรับการปรับแต่ง (Refinement) แต่ในขณะเดียวกันตัวป้องกันในอีเวนท์บีมีประสิทธิภาพสำหรับการปรับแต่ง ซึ่งสามารถช่วยให้ผู้ใช้สร้างโมเดลได้เหมือนจริงมากยิ่งขึ้น

คุณลักษณะสำคัญของอีเวนท์บี คือการสร้างแบบจำลองเชิงรูปนัยสำหรับการทวนสอบและการวิเคราะห์ระบบที่เกิดการทำงานพร้อมกันที่เน้นความปลอดภัย (Safety Critical System) โดยใช้ทฤษฎีเซตสำหรับสร้างแบบจำลองเพื่อสามารถใช้การพิสูจน์ทางคณิตศาสตร์ (Mathematical Proof) สำหรับการตรวจสอบความสอดคล้อง (Consistency) และความถูกต้อง (Correctness) ของข้อมูลภายในระบบซอฟต์แวร์ โครงสร้างของอีเวนท์บีแบ่งเป็นสองส่วนประกอบคือส่วนประกอบที่เป็นนามธรรม (Abstract Component) เป็นการระบุข้อกำหนดความต้องการเริ่มต้นในโครงสร้างพื้นฐานในส่วนประกอบของ บริบท (Context) ส่วนประกอบต่อมาคือส่วนที่จำลองพฤติกรรมระบบที่ขับเคลื่อนการทำงานด้วยเหตุการณ์ (Event-driven) ในส่วนประกอบของแมชชีน (Machine) และมีการตรวจสอบการการปรับแต่ง (Refinement) ของข้อกำหนดที่เป็นนามธรรม (High level) ให้อยู่ในรูปแบบของรูปธรรมได้ (Low level) แต่ละขั้นตอนของการปรับแต่งจะได้รับการพิสูจน์ตามข้อกำหนดที่ระบุไว้เพื่อตรวจสอบความสอดคล้องและความถูกต้องของข้อกำหนดที่เป็นนามธรรม เรียกว่า การพิสูจน์ข้อผูกพัน (Proof Obligations) สำหรับการพิสูจน์สามารถทำได้อัตโนมัติโดยใช้เครื่องมือที่พัฒนาพร้อมกับอีเวนท์บี คือเครื่องมือชื่อว่า โรดิน (Rodin) ใช้สร้างแบบจำลองโดยอีเวนท์บีและสามารถระบุข้อกำหนดเพื่อตรวจสอบความสอดคล้องและความถูกต้องของแบบจำลองเมื่อเกิดการเปลี่ยนแปลงได้อย่างอัตโนมัติ

2.1.2.1 ส่วนประกอบของอีเวนท์บี (Component of Event-B)

อีเวนท์บีถูกแบ่งออกเป็นสองส่วนประกอบคือ บริบท (Contexts) เป็นส่วนที่แสดงพฤติกรรมแบบสถิต (Static behavior) และพฤติกรรมแบบแบบพลวัต (Dynamic behavior) จะแสดงในส่วนประกอบแมชชีน (Machine) มีรายละเอียด ดังต่อไปนี้

1) บริบท (Context)

บริบทเป็นส่วนของการระบุข้อกำหนดแบบคงที่ของระบบซึ่งจะไม่ถูกเปลี่ยนแปลงค่าเมื่อเวลาผ่านไป ประกอบด้วยชุดผู้ให้บริการ, ค่าคงที่, สัจพจน์และทฤษฎีบท บริบทสามารถขยาย (Extend) ได้จากบริบทอื่นได้ ข้อกำหนดถูกเรียกใช้โดยแมชชีนเพื่อเป็นข้อกำหนดพฤติกรรมระบบ แต่ไม่ยอมให้มีการเปลี่ยนแปลงค่าของบริบท แสดงคำอธิบายในตารางที่ 2.4

ตารางที่ 2.4 องค์ประกอบของบริบท [7]

องค์ประกอบ	คำอธิบาย
Context c_Context_Name	บริบท (Context) คือ การกำหนดให้บริบทมีชื่อเรียกว่า “c_Context_Name”
Set (s)	ชุดผู้ให้บริการ (Carrier sets) คือ ชุดของผู้ให้บริการที่ประกาศตัวตนด้วยชื่อ แต่ละบริบทชุดผู้ให้บริการจะมีความอิสระต่อกันและไม่สามารถมีค่าว่างได้
Constant (l)	ค่าคงที่ (Constants) คือ ค่าถูกกำหนดโดยชุดของการทำนาย (Set of Predicates) เพื่อใช้สำหรับการทำนายว่าระบบตรงตามคุณสมบัติหรือไม่
Axioms (s, c)	สัจพจน์ (Axioms) คือ กฎที่นิยามขึ้นที่เป็นชุดของการทำนาย (Set of Predicates) สำหรับทำนายคุณสมบัติของชุดผู้ให้บริการและค่าคงที่ ชุดการทำนายเหล่านี้สามารถนำกลับมาใช้ใหม่ได้เมื่อมีการสร้างแบบจำลองไม่ต้องได้รับการพิสูจน์ใหม่อีกครั้ง
Theorems	ทฤษฎีบท (Theorems) คือ คุณสมบัติของระบบสามารถพิสูจน์ได้โดยสัจพจน์

2) แมชชีน (Machine)

แมชชีนเป็นส่วนที่อธิบายพฤติกรรมแบบพลวัตรของระบบที่ทำงานพร้อมกัน แมชชีนสามารถถูกปรับแต่ง (Refinement) จากแมชชีนอื่นได้และสามารถมองเห็นค่าที่อยู่ในบริบทสำหรับเรียกใช้เพื่อขับเคลื่อนอีเวนท์ที่อยู่ภายในแมชชีนได้ ซึ่งลักษณะสำคัญของการสร้างแมชชีนคือการพิสูจน์ว่าอีเวนท์ทั้งหมดที่อยู่ในแมชชีนไม่อยู่ในสถานะที่ไม่ถูกต้อง มีองค์ประกอบดัง

ตารางที่ 2.5

ตารางที่ 2.5 องค์ประกอบของแมชชีน [7]

องค์ประกอบ	คำอธิบาย
Machine m_Machine_Name	แมชชีน (Machine) คือ การกำหนดให้แมชชีนมีชื่อเรียกว่า “m_Machine_Name”
Variables (v)	ตัวแปร (Variables) คือ ตัวแปรที่ประกาศขึ้นโดยไม่มีการกำหนดค่าและประเภทของตัวแปร
Invariants I(v)	ตัวยืนยง (Invariants) คือ ค่าคงที่ของตัวแปรที่ถูกประกาศ โดยจะกำหนดประเภทของตัวแปรที่ถูกประกาศขึ้นก่อนหน้า
Events	เหตุการณ์ (Events) คือ เหตุการณ์ที่อาจจะเกิดขึ้นภายในระบบ เหตุการณ์จะเกิดได้ก็ต่อเมื่อเงื่อนไข (Guard) เป็นจริงเท่านั้น
Proof Obligations	การพิสูจน์ข้อผูกพันถูกสร้างและดำเนินการพิสูจน์อย่างอัตโนมัติตามพฤติกรรมที่อยู่ในแมชชีน

3) อีเวนท์ (Events)

อีเวนท์หรือเหตุการณ์คือตัวจำลองการเกิดพฤติกรรมระบบซึ่งประกอบด้วยเหตุการณ์เริ่มต้น (Initialisation) ที่เป็นเหตุการณ์เกิดเป็นอันดับแรก สำหรับระบุค่าเริ่มต้นให้กับตัวแปรที่ถูกประกาศไว้ก่อนหน้าก่อนที่จะเกิดเหตุการณ์อื่นและเหตุการณ์ที่อาจจะเกิดขึ้น ประกอบด้วยด้วยเงื่อนไข (Guard) และดำเนินการอธิบายสถานะของการเปลี่ยนแปลงเมื่อเงื่อนไขเป็นจริง ตารางที่ 2.6 แสดงรูปแบบทั่วไปของอีเวนท์

ตารางที่ 2.6 รูปแบบทั่วไปของอีเวนท์ [7]

รูปแบบของอีเวนท์	คำอธิบาย
event Initialisation **then** gb := 10 **end**	เหตุการณ์เริ่มต้น กำหนดให้ตัวแปร (gb) มีค่าเท่ากับ 10
event event_name **any** b **where** G(b,gb) **then** GS(b,gb)	ใช้ตัวแปรส่งเข้าเพื่อตรวจสอบเงื่อนไข any b คือ ตัวแปรที่ใช้ในการตรวจสอบเงื่อนไง where G(b, gb) คือ การตรวจสอบตัวแปร (b) สามารถทำให้เงื่อนไขเป็นจริงภายใต้ตัวแปรทั่วไป (gb) หรือไม่ then GS(b, gb) เมื่อเงื่อนไขเป็นจริง (gb) จะเปลี่ยนสถานะโดยการแทนค่าด้วย (b)

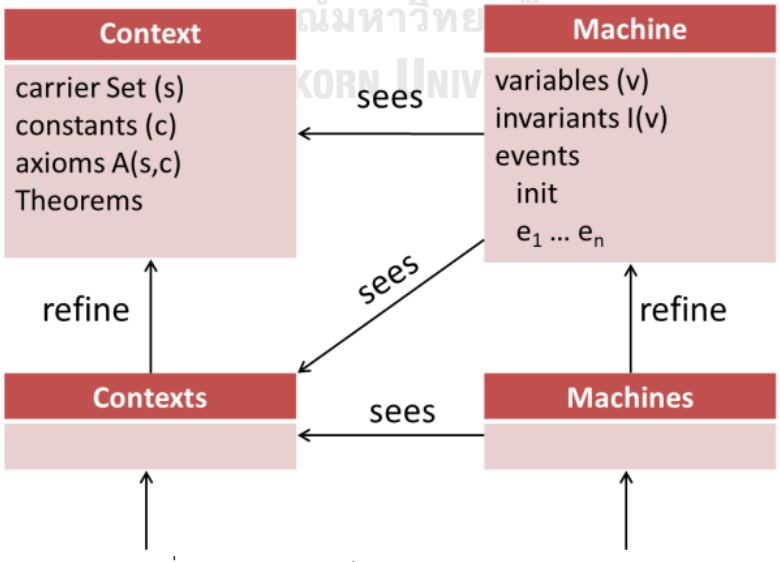
รูปแบบของอีเวนท์	คำอธิบาย
end	

ตารางที่ 2.6 รูปแบบทั่วไปของอีเวนท์ [7] (ต่อ)

รูปแบบของอีเวนท์	คำอธิบาย
event event_name when G(gb) then GS(gb,gb') end	ใช้ตัวแปรทั่วไปที่อยู่ในแมชชีนเป็นตัวตรวจสอบเงื่อนไข เมื่อเงื่อนไขเป็นจริง จะถูกเปลี่ยนสถานะของตัวแปร (gb) ด้วย (gb')
event event_name begin GS(gb,gb') end	ถ้าเงื่อนไขเป็นจริงเสมอ เปลี่ยนสถานะของตัวแปร (gb) ด้วย (gb')

4) ความสัมพันธ์ระหว่างบริบทและแมชชีน (Relationship between context and machine)

บริบทและแมชชีนมีความสัมพันธ์กันโดยที่แมชชีนสามารถถูกเปลี่ยนแปลง (Refinement) ได้จากแมชชีนอื่น นอกจากนี้แมชชีนยังสามารถที่จะมองเห็นหลายๆ บริบทได้ และบริบทเป็นส่วนขยายให้ผู้อื่นได้ แต่ไม่อนุญาตให้เกิดความสัมพันธ์ที่เป็นวงจรแสดงความสัมพันธ์ระหว่างบริบทและแมชชีน ดังรูปที่ 2.4



รูปที่ 2.4 ความสัมพันธ์ระหว่างบริบทและแมชชีน [7]

5) สัญกรณ์ทางคณิตศาสตร์ของอีเวนท์บี (Notation of mathematics in Event-B)

การทวนสอบเชิงรูปนัยได้นำทฤษฎีเซตในเชิงทางคณิตศาสตร์เข้ามาประยุกต์เพื่อทำการทวนสอบแบบจำลองต่างๆ เซตในทางคณิตศาสตร์หมายถึงกลุ่มของวัตถุที่อยู่รวมกัน ซึ่งเป็นพื้นทางของคณิตศาสตร์ยุคใหม่ได้แก่ ตรรกศาสตร์และแคลคูลัส สมาชิกที่อยู่ภายในเซตสามารถเป็นอะไรก็ได้ เช่น ตัวเลข ตัวอักษร กลุ่มคำ หรือเซตอื่นๆ การกำหนดทฤษฎีเซตมีได้รับความน่าเชื่อถือโดยทั่วไปและถูกนำมาประยุกต์ใช้ได้กับทุกสาขาวิชา อีเวนท์บีได้นำทฤษฎีเซตเข้ามาประยุกต์เพื่อพิสูจน์ความถูกต้องของการทำงาน แสดงตัวอย่างในตารางที่ 2.7

ตารางที่ 2.7 ตัวอย่างสัญกรณ์ทางคณิตศาสตร์ของอีเวนท์บี [7]

ความหมาย	สัญกรณ์ของอีเวนท์บี	แอสกี (ASCII)
จริง (TRUE)	$\top$	TRUE
เท็จ (FALSE)	$\bot$	FALSE
ประพจน์เชื่อม (Conjunction)	P $\cup$ Q	P & Q
ประพจน์เลือก (Disjunction)	P $\cap$ Q	P or Q
ประพจน์มีเงื่อนไข (Implication)	P => Q	P => Q
นิเสธ (Negation)	-P	not P
ตัวบ่งปริมาณทั้งหมด (Universal quantification)	$\forall$ x1, ... xn • P(x1, ... xn)	!z.P => Q
ตัวบ่งปริมาณมีอย่างน้อยหนึ่ง (Existential quantification)	$\exists$ x1, ... xn • P(x1, ... xn)	#z.P & Q
จำนวนเต็ม (Integer)	$\mathbb{Z}$	INT
จำนวนเต็มบวก (Positive Number)	N1	NAT1
เพาเวอร์เซต (Power set)	P	POW

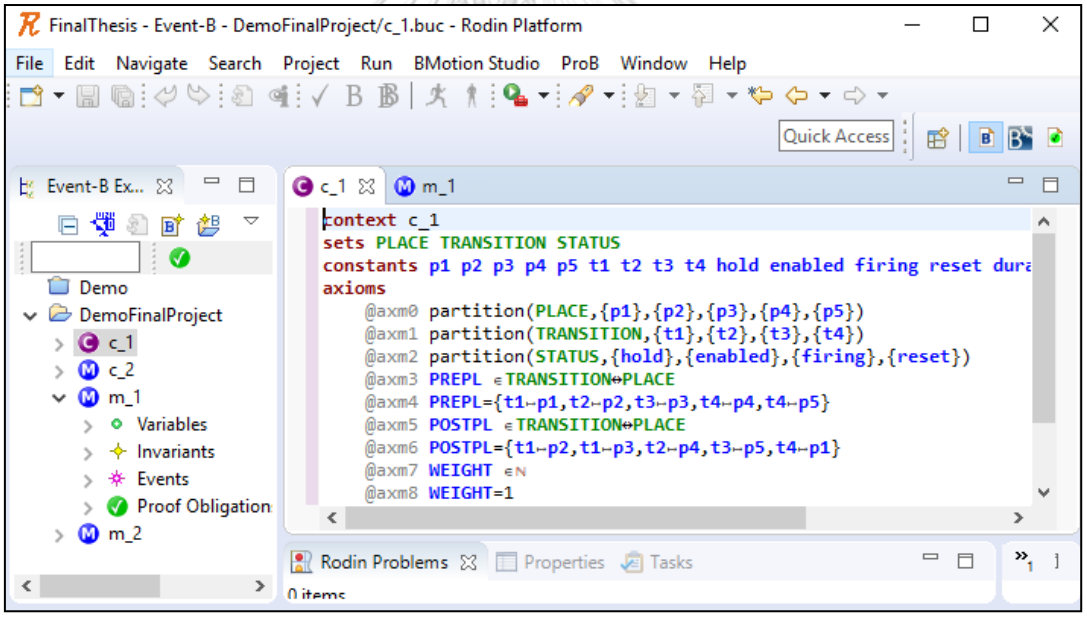
2.1.2.2 เครื่องมือโรดิน (Rodin tool)

เครื่องมือโรดิน [11] มีชื่อเต็มคือสภาพแวดล้อมของการพัฒนาที่แม่นยำสำหรับระบบที่ซับซ้อน (Rigorous Open Development Environment for Complex Systems) ถูกพัฒนาพร้อมกับอีเวนท์บี เพื่อจำลองโมเดลเชิงรูปนัยของอีเวนท์บี โดยมีคุณลักษณะสำคัญคือ ความสะดวกสบายในการสร้างโมเดลอีเวนท์บีและสนับสนุนการเพิ่มส่วนต่อขยายได้ โรดินสนับสนุนการปรับแต่งและการพิสูจน์ทางคณิตศาสตร์เพื่อเพื่อพิสูจน์ความถูกต้องของแมชชีนได้อย่างอัตโนมัติ โรดินถูกพัฒนาโดยมีพื้นฐานมาจากเครื่องมือพัฒนา

โปรแกรมอีคิปส์ด้วยภาษาจาวา โรดินมีความยืดหยุ่นสำหรับการเพิ่มส่วนต่อขยายเข้าเพื่อเพิ่มความสามารถของเครื่องมือโรดินมากขึ้น เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียนว่าส่วนต่อขยาย (Plug-in) มีทั้งหมด 7 ประเภท มีดังนี้

- การสร้างแบบจำลอง (Modeling)
- การทำภาพเคลื่อนไหว (Animation)
- การมองเห็น (Visualization)
- เอกสาร (Documentation)
- ทฤษฎีและการพิสูจน์ (Theory and Proof)
- การสร้างโค้ด (Code Generation)
- การทดลอง (Experimental)

งานวิจัยนี้ได้ใช้ส่วนต่อขยายประเภททฤษฎีและการพิสูจน์ได้แก่เอสเอ็มที (SMT) และโปรบี (ProB) เพื่อเพิ่มประสิทธิภาพในการพิสูจน์ความสอดคล้องและความถูกต้องของเครื่องมือโรดินและตรวจสอบคุณสมบัติด้านความปลอดภัยของอีเวนท์บีแสดงตัวอย่างในรูปที่ 2.5



รูปที่ 2.5 ตัวอย่างเครื่องมือโรดิน

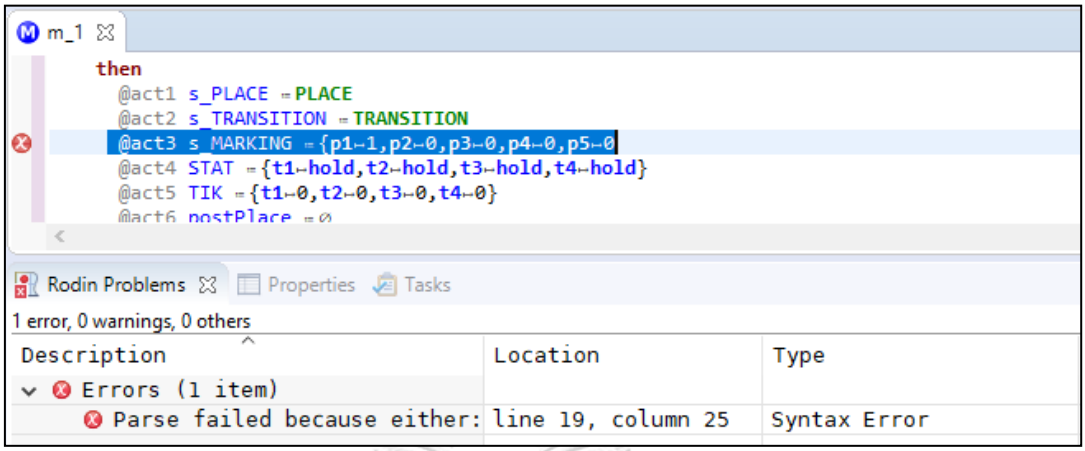
2.1.2.3 การพิสูจน์ (Proving)

เครื่องมือโรดินมีความสามารถในการพิสูจน์อย่างอัตโนมัติประกอบด้วย 3 ส่วน ดังนี้

1) การตรวจสอบแบบสถิต (Static check)

การตรวจสอบแบบสถิต (Static check) เป็นการวิเคราะห์คำศัพท์ (Lexical analysis), วิเคราะห์ประโยค (Syntactic analysis) และตรวจสอบแบบชนิด (Type check) หากการตรวจสอบแบบสถิตตรวจ

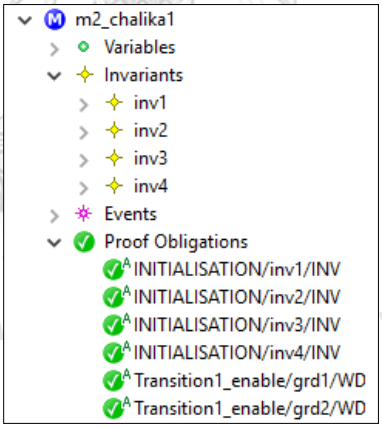
พบความผิดพลาดในขณะที่พัฒนาจะแสดงในหน้าต่าง “Rodin Problem” ซึ่งโปรแกรมจะไม่สามารถรันได้ หากความผิดพลาดได้รับการแก้ไข แสดงตัวอย่างดังรูปที่ 2.6



รูปที่ 2.6 ผลการตรวจสอบแบบสถิต

2) ตัวสร้างข้อผูกพัน (Proof obligation generator)

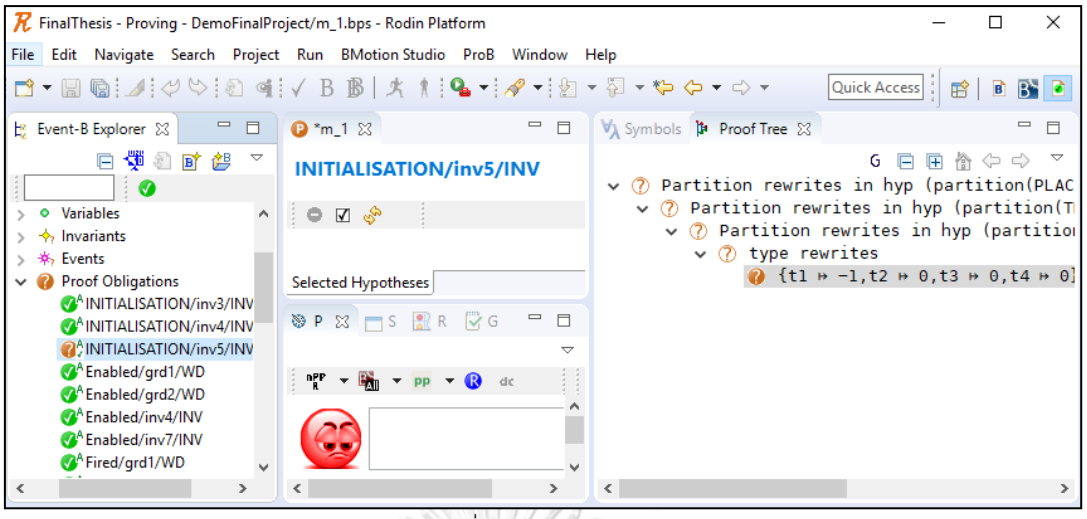
ตัวสร้างข้อผูกพัน (Proof obligation generator) ทำหน้าที่สร้างข้อผูกพันจากบริบทหรือแมชชีนอย่างอัตโนมัติโดยจะสร้างจากเนื้อความที่อยู่ในแมชชีนที่ละบรรทัด และแสดงในลำดับชั้นของ “Proof Obligations” ภายใต้แมชชีนเป้าหมาย แสดงตัวอย่างดังรูปที่ 2.7



รูปที่ 2.7 ผลการสร้างข้อผูกพัน

3) ผู้พิสูจน์ (Prover)

ผู้พิสูจน์ทำหน้าที่นำข้อผูกพันที่สร้างมาพิสูจน์เพื่อตรวจสอบความถูกต้องของบริบทหรือแมชชีนอย่างอัตโนมัติ การพิสูจน์จะดำเนินครบทุกบรรทัดตามที่ผู้สร้างข้อผูกพันทางหลักฐานสร้างออกมาก ผลการพิสูจน์หากไม่พบข้อผิดพลาดหรือช่องโหว่ของบริบทหรือแมชชีนจะแสดงสัญลักษณ์ตามรูปที่เป็นสีเขียว แต่ถ้าตรวจพบข้อผิดพลาดหรือช่องโหว่ของบริบทหรือแมชชีนจะแสดงสัญลักษณ์ตามรูปที่เป็นสีแดงจากรูปที่ 2.8 แสดงให้เห็นว่ามีข้อผิดพลาดของการพิสูจน์เกิดขึ้นของตัวยืนยง 5 ในอีเวนท์ “Intialisation”



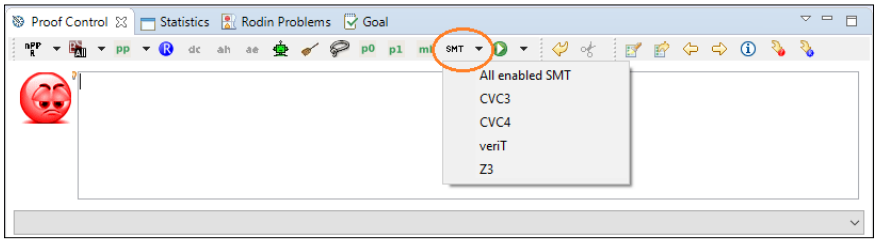
รูปที่ 2.8 ผลการพิสูจน์

เครื่องมือโรดินมีความสามารถในการพิสูจน์ได้อย่างอัตโนมัติแต่ก็ยังมีข้อจำกัดที่ไม่สามารถพิสูจน์ได้อย่างอัตโนมัติเพราะกระบวนการพิสูจน์ไม่ได้รับสัจพจน์หรือข้อกำหนดไม่สมบูรณ์ ซึ่งในบางกรณีสามารถแก้ไขได้ตามคำแนะนำให้เพิ่มเติมกฎหรือข้อกำหนดเพื่อให้สามารถดำเนินการพิสูจน์ได้ แต่การเพิ่มเติมจะไม่สามารถทำได้อย่างอัตโนมัติทำให้เสียเวลา เพราะบางกรณีที่ผลการพิสูจน์ผิดพลาดเกิดจากการได้รับข้อกำหนดไม่สมบูรณ์แต่ความจริงผลการพิสูจน์จะไม่เกิดข้อผิดพลาด

ดังนั้นจึงมีการพัฒนาเครื่องมือพิสูจน์ที่อยู่ในรูปแบบของส่วนต่อขยาย (Plug-in) ที่ทำให้โรดินมีประสิทธิภาพมากยิ่งขึ้น ในงานวิจัยนี้ได้ใช้ส่วนต่อขยายประเภททฤษฎีและการพิสูจน์ทั้งหมดสองตัวคือส่วนต่อขยายเอสเอ็มทีและส่วนต่อขยายโปรบี อธิบายรายละเอียดในหัวข้อย่อยต่อไป

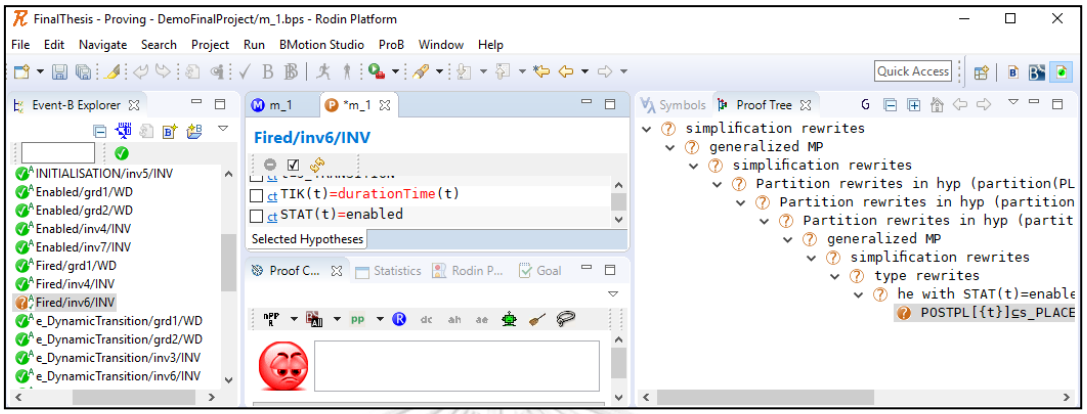
4) ส่วนต่อขยายเอสเอ็มที (SMT plug in)

เอสเอ็มที [12] มีชื่อเต็มคือทฤษฎีความแซดทิสอะบิลิตี้โมดูโล (Satisfiability Modulo Theory) เอสเอ็มทีได้รับความนิยมและได้รับการทดสอบจากผู้ที่สามารถแก้ปัญหาคือ ซีวีซี3 (CVC3), เวอร์ริที (veriT) [13], แซด3 (Z3) และแอลทเออโก (Alt-Ergo) [14] ทั้ง 4 ตัวได้ถูกรวมอยู่ในเอสเอ็มทีและถูกใช้อย่างเป็นทางการในการค้นหาแหล่งที่มาในเครื่องมือโรดิน เมื่อติดตั้งจะปรากฏอยู่ในหน้าต่าง “Proof Control” ดังรูปที่ 2.9

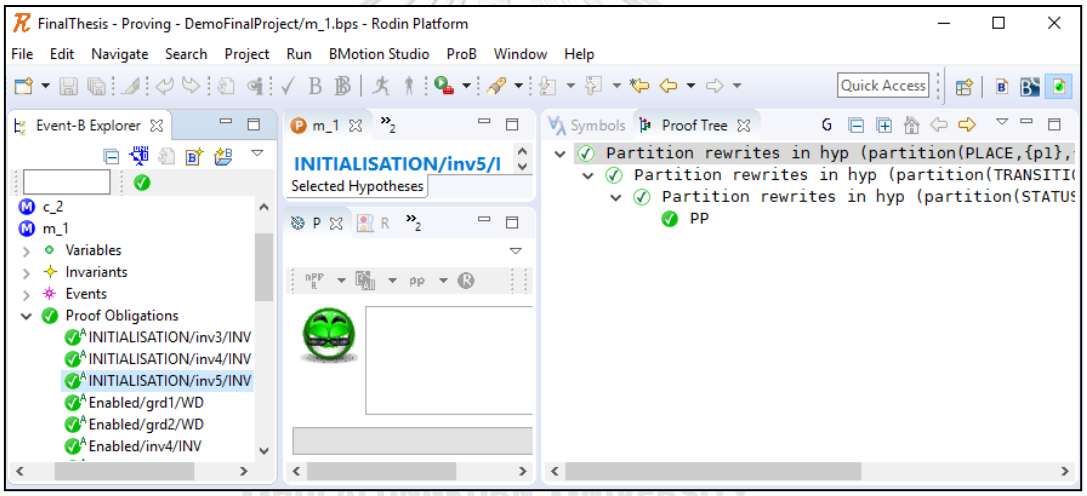


รูปที่ 2.9 ส่วนต่อขยายทฤษฎีแซดทิสอะบิลิตี้โมดูโล

เอสเอ็มทีได้นับการทดสอบกับโครงการยูโรเปียน (European project) [15] เพื่อหาความสามารถการพิสูจน์ข้อผูกพัน ซึ่งผลการทดลองแสดงให้เห็นว่าอัตราส่วนของการพิสูจน์เพิ่มขึ้นจากเดิม 53% ไปเป็น 63% ดังนั้นการพิสูจน์จะมี 2 ขั้นตอนคือ ขั้นตอนการพิสูจน์ของโรดินหากผลการพิสูจน์ของโรดินยังเกิดขึ้นผิดพลาดจึงจะดำเนินการพิสูจน์ของเอสเอ็มทีอีกครั้ง แสดงตัวอย่างดัง รูปที่ 2.10 และ รูปที่ 2.11



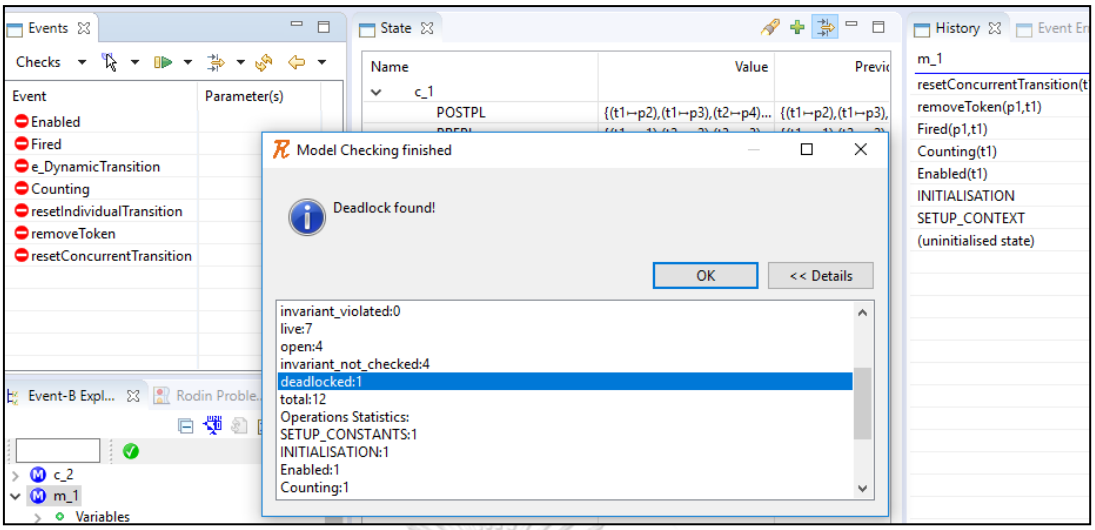
รูปที่ 2.10 ตัวอย่างการพิสูจน์ข้อผูกพันโดยเครื่องมือโรดิน



รูปที่ 2.11 ตัวอย่างการพิสูจน์ข้อผูกพันโดยส่วนต่อขยายเอสเอ็มที

5) ส่วนต่อขยายโปรบี (ProB plug in)

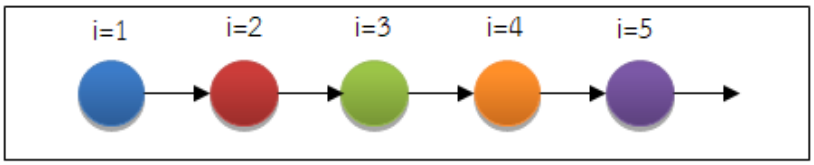
โปรบี [16] คือเครื่องมือที่สร้างภาพที่เคลื่อนไหว (Animator) แก้ไข้ข้อจำกัดของการทวนสอบและตรวจสอบความถูกต้องของบีเมทอด (B-Method) อีกทั้งยังสนับสนุนอีเวนท์บี เพื่อการทวนสอบข้อกำหนดแบบละเอียดเพื่อหาข้อผิดพลาดสามารถตรวจสอบภาวะติดตาย (Deadlocks testing) และสร้างกรณีทดสอบ โปรบีสามารถติดตั้งในเครื่องมือโรดินในรูปแบบส่วนต่อขยาย ที่มาพร้อมกับบีโมชันสตูดิโอ (BMotionStudio) เพื่อสร้างกราฟิกของโดเมนเฉพาะของโมเดล ซึ่งจะจำลองการเคลื่อนที่ของเหตุการณ์ที่เกิดขึ้นภายในแมชชีน นอกจากนั้นโปรบีสนับสนุนการใช้ตรรกศาสตร์เวลาแบบเชิงเส้น (Linear Temporal Logic : LTL) สำหรับตรวจสอบคุณสมบัติของอีเวนท์บี ตัวอย่างการทวนสอบโดยใช้ส่วนต่อขยายโปรบี แสดงดังรูปที่ 2.12



รูปที่ 2.12 ตัวอย่างการใช้เครื่องมือโปรบีสำหรับการตรวจสอบภาวะติดตาย

2.1.2.4 ตรรกศาสตร์เชิงกาลเวลา (Temporal Logic/ Liner Temporal Logic)

ในปี 1977 Amir Pnueli [17] ได้นำเสนอตรรกศาสตร์เชิงกาลเวลาสำหรับการทวนสอบเชิงรูปนัยของชุดคำสั่งคอมพิวเตอร์ โดยการสร้างประโยคตรรกศาสตร์จากสัญลักษณ์ที่แสดงข้อกำหนดของคุณสมบัติของระบบภายใต้เงื่อนไขของเวลา เช่น “ถ้าตัวแปร ‘เอ’ เป็นจริงแล้ว เมื่อเวลาผ่านไปเรื่อยๆแล้ว ในที่สุดแล้ว ‘เอ’ ก็ยังเป็นจริง”คุณลักษณะสำคัญของตรรกศาสตร์เชิงเวลาคือสามารถอธิบายโครงสร้างและพฤติกรรมของซอฟต์แวร์และฮาร์ดแวร์ได้อย่างสมจริงและสามารถทวนสอบเชิงรูปนัยแบบโมเดลเช็คกิงสำหรับทวนสอบคุณสมบัติของการทำงานทั้งหมดของระบบตรงตามข้อกำหนดเชิงรูปนัย รูปแบบของประโยคตรรกศาสตร์เชิงกาลเวลาอยู่ในรูปแบบของลำดับของสถานะที่เป็นอนันต์ การตรวจสอบทำตามลำดับของสถานะที่ระบุด้วยดัชนี (i = 0, 1, 2,..) แสดงตัวอย่างในรูปที่ 2.13



รูปที่ 2.13 ตรรกศาสตร์เชิงเวลา

1) ไวยากรณ์และความหมาย (Syntax and Semantic)

ตรรกศาสตร์เชิงกาลเวลาประกอบด้วย ประพจน์เดี่ยว (Atomic Propositions) ตัวดำเนินการบูลีน (Boolean Operators) และตัวดำเนินการเชิงเวลา (Temporal Operators) อธิบายความหมายของตัวดำเนินการชั่วคราวในได้ดังนี้

- a หมายความว่า a ประพจน์เดี่ยวที่มีค่าความจริงเป็นจริง ณ ตอนนี้
- X a หมายความว่า a เป็นจริงในสถานะถัดไป (Next)
- F a หมายความว่า a จะเป็นจริงในอนาคต (Future)

- G a หมายความว่า a จะเป็นจริงในอนาคตตลอดไป (Globally)
- a U b หมายความว่า a จะเป็นจริงจนกระทั่ง b เป็นจริง

2) คุณสมบัติของตรรกะเชิงเวลา (Linear Temporal properties)

- ความปลอดภัย (Safety) คือพฤติกรรมของระบบจะไม่มีเหตุการณ์ไม่ดีเกิดขึ้น

ตัวอย่าง ยอดเงินในบัญชีธนาคารจะต้องไม่มีค่าน้อยกว่าศูนย์ตลอดไป

แทนด้วย G (ยอดเงินในบัญชีมีค่ามากกว่าหรือเท่ากับศูนย์)

- ความคงอยู่ (Liveness) คือพฤติกรรมของระบบจะมีสิ่งที่ดีเกิดขึ้น

ตัวอย่าง เมื่อมีคำร้องขอเกิดขึ้น ในที่สุดแล้วจะมีการตอบรับทราบ

แทนด้วย G (คำร้องขอ => ตอบรับทราบ)

- ความเป็นธรรม (Fairness) คือถ้าเกิดเหตุการณ์ขึ้นจะมีอีกเหตุการณ์เกิดขึ้นตามเป็นเหตุผลรองรับ

ตัวอย่าง ถ้าข้อความถูกส่งไปหาผู้รับแล้วข้อความจะได้รับในที่สุด

แทนด้วย ส่งข้อความ => F (รับข้อความ)

2.2 งานวิจัยที่เกี่ยวข้อง

เมื่อได้ศึกษาทฤษฎีที่เกี่ยวข้องแล้วผู้วิจัยได้ศึกษาค้นคว้างานวิจัยที่เกี่ยวข้องกับงานวิจัยได้เลือกนำเสนอทั้งหมด 5 งานวิจัยที่มีความสอดคล้องกับงานวิจัยครั้งนี้ อธิบายในหัวข้อย่อยดังต่อไปนี้

2.2.1 งานวิจัยชื่อ Semantic Embedding of Petri Nets into Event-B โดย Christian Attiogbe ปี ค.ศ. 2005

งานวิจัยนี้ [18] ได้นำเสนอนำเสนอการฝังวิธีเชิงรูปนัยของเพทริเน็ตไปเป็นระบบนามธรรมบีสำหรับการสร้างแบบจำลองของระบบแบบทำงานพร้อมกัน ผู้วิจัยใช้เทคนิคการฝัง (Embed Techniques) แบบการฝังความหมาย (Semantic Embedding) เป็นเทคนิคที่ใช้กันอย่างแพร่หลายสำหรับการรวมวิธีการและสัญลักษณ์แบบอัตโนมัติ สามารถให้นำระเบียบวิธีการนำกลับมาใช้ใหม่ของกรอบทางตรรกะที่มีอยู่สำหรับการวิเคราะห์เชิงรูปนัยเหมือนกัน มีขั้นตอนของระบบนามธรรมต่อไปนี้

ขั้นตอนที่ 1 การฝังโครงสร้างเพทริเน็ตภายในบี (Embedding the Structure of Petri Nets within B)

ขั้นตอนที่ 2 การฝังความหมายของเพทริเน็ตไปยังบี (Embedding Petri Nets Evolution Semantics into B)

ขั้นตอนที่ 3 การฝังเพทริเน็ตที่มีการดำเนินการที่แนบกับเพลสลงในบี (Embedding into B of Petri Nets with Actions Attached to Places)

ขั้นตอนที่ 4 การฝังเพทริเน็ตที่มีการดำเนินการที่เกี่ยวข้องกับทรานซิชัน (Embedding into B of Petri Nets with Actions Attached to Transitions)

ขั้นตอนที่ 5 การฝังเพทริเน็ตที่มีการดำเนินการที่เกี่ยวข้องกับเพลสและทรานซิชัน (Embedding into B of Petri Nets with Actions Attached to both Places and Transitions)

ขั้นตอนทั้งหมดเป็นการจำลองเหตุการณ์ที่มีโอกาสเกิดขึ้นตามลำดับความง่ายไปยาก เพื่อหลีกเลี่ยงพฤติกรรมที่ไม่ถูกต้อง แต่ละขั้นตอนจะได้ผลของฝังในรูปแบบของบี เมื่อได้ผลการจำลองระบบจากเพทริเน็ตไปยังบี ผู้วิจัยใช้การตรวจสอบความสอดคล้องของเพทริเน็ตและบีโดยใช้เครื่องมือแอทเลอร์บี (Atelier B) และวิเคราะห์ปัญหาด้วยกรณีศึกษาที่มีขนาดเล็กหลายกรณี งานวิจัยนี้ไม่ได้แปลงเพทริเน็ตไปยังอีเวนท์บีโดยตรง เป็นการใช้นามธรรมของบีที่เป็นพื้นฐานของบีเมทอดและบีเมทอดผู้วิจัยทดลองใช้กฎการแปลงที่ได้จากผลลัพธ์จาก 5 ขั้นตอนมาปรับเปลี่ยนให้เป็นอีเวนท์บี ซึ่งโครงสร้างที่เป็นอีเวนท์บีไม่ได้รับการตรวจสอบความถูกต้องในการฝังความหมายและยังไม่มีเครื่องมืออัตโนมัติสำหรับการฝังเพทริเน็ตไปยังนามธรรมบี

งานวิจัยดังกล่าวมีความแตกต่างกับงานของผู้วิจัย โดยที่งานวิจัยดังกล่าวจากเพทริเน็ตแบบดั้งเดิมเป็นบีเมทอดจากนั้นและใช้วิธีเทคนิคการฝังความหมายของบีเมทอดแปลงไปยังอีเวนท์บี ซึ่งเป็นเพียงแนวคิดของในของระดับของนามธรรมและยังไม่ได้รับการตรวจสอบความถูกต้องของผลลัพธ์การแปลง ดังนั้นผู้วิจัยได้ศึกษางานวิจัยได้กล่าวเพื่อทำความเข้าใจลักษณะโครงสร้างของบีเมทอดซึ่งมีความคล้ายกับอีเวนท์บีเพื่อให้ได้ผลลัพธ์ที่สามารถนำไปประมวลผลตรวจสอบความสอดคล้องบนเครื่องมือโรดิน

2.2.2 งานวิจัยชื่อ The EventB2PN Tool: From Event-B Specification to Petri Nets through Model Transformation โดย Mohamed Garoui, Belhassen Mazigh และ Abderrafiaa Koukam ปี ค.ศ. 2015

งานวิจัยนี้ [19] ได้นำเสนอเครื่องมือการแปลงอีเวนท์บีไปเป็นเพทริเน็ตเพื่อหาปริมาณของการตรวจสอบหรือการวิเคราะห์คุณลักษณะต่างๆ เช่น การปริมาณความน่าเชื่อถือของระบบหรือปริมาณความพร้อมใช้งานของระบบ ดังนั้นวิจัยนี้ได้ออกแบบกฎการแปลงอีเวนท์บีไปเป็นเพทริเน็ตและพัฒนาเครื่องมือสนับสนุนการแปลงชื่อว่า “EventB2PN” เป็นเครื่องมือสนับสนุนการแปลงอีเวนท์บีไปเป็นเพทริเน็ตเพื่อสนับสนุนการวิเคราะห์ประสิทธิภาพเชิงปริมาณของอีเวนท์บี การวัดประสิทธิภาพจะกำหนดสิ่งที่คำนวณระหว่างการวิเคราะห์ ค่าปกติจะเป็นจำนวนเต็มเฉลี่ยในแต่ละเพลส เมื่อได้รับผลลัพธ์การแปลงจากอีเวนท์บีเป็นเพทริเน็ตจากเครื่องมือที่พัฒนาขึ้นมาจะได้รับการประเมินผลลัพธ์โดยเครื่องมือไทมด์เน็ต (TimeNET) เป็นเครื่องมือซอฟต์แวร์สำหรับการสร้างแบบจำลองและการประเมินผลการปฏิบัติงานสำหรับเพทริเน็ต กฎการแปลงของงานวิจัยนี้นำเสนอออกมาทั้งหมดสามกฎดังต่อไปนี้

กฎข้อที่ 1 จากตัวแปรและค่าคงเป็นเพลสและสเตทสเปซ (From Variables and Invariants To PN places and State Space)

กฎข้อที่ 2 จากอีเวนท์การเริ่มต้นเป็นการกำหนดค่าเริ่มต้นของมาร์กกิง (From INITIALISATION To PN Initial Marking)

กฎข้อที่ 3 จากอีเวนท์เป็นทรานซิชัน (From Events To PN Transitions)

งานวิจัยดังกล่าวงานวิจัยมีจุดประสงค์ตรงกันข้ามกับงานของผู้วิจัยคือการพัฒนาเครื่องมือสำหรับเพิ่มประสิทธิภาพของการวิเคราะห์คุณลักษณะเชิงปริมาณของอีเวนท์บี โดยการแปลงอีเวนท์บีไปเป็นเพทริเน็ต แต่สำหรับงานของผู้วิจัยมีจุดประสงค์การแปลงไทมด์เพทริเน็ตเป็นอีเวนท์บี แต่งานวิจัยดังกล่าวเป็นมีประโยชน์สำหรับแนวคิดเกี่ยวกับความสัมพันธ์เชิงโครงสร้างของอีเวนท์บีและเพทริเน็ตเพื่อออกแบบกฎการแปลงไทมด์เพทริเน็ตเป็นอีเวนท์บีเพื่อให้ได้ผลลัพธ์ที่สมบูรณ์แบบและไม่เกิดข้อผิดพลาดในการดำเนินงานวิจัย

2.2.3 งานวิจัยชื่อ Petri Nets to B-Language Transformation in Software Development โดย Stefan Korecko และ Branislav Sobota ปี ค.ศ. 2014

งานวิจัยนี้ [20] ได้นำเสนอนำเสนอการแปลงเพทริเน็ตไปเป็นภาษาบีเมทอดในการพัฒนาซอฟต์แวร์ของระบบแบบเป็นลำดับ (Sequential System) และได้พัฒนาซอฟต์แวร์ที่ใช้ในการทดลองนี้ ชื่อว่า “mFDTE/PNtool2 software” งานวิจัยนี้ได้แสดงให้เห็นความสัมพันธ์ระหว่างเพทริเน็ตและบีเมทอด เป็นแนวทางการกำหนดสัญลักษณ์ต่างๆ ของเพทริเน็ตแปลงเป็นภาษาบีเมทอด และอธิบายความหมายสำหรับการแทนโดยรวม (Generalized Substitutions) ซึ่งถูกใช้เป็นเงื่อนไขของพฤติกรรมของระบบ งานวิจัยนี้นำเสนอกฎในการแปลงเพทริเน็ต ไปเป็นบีเมทอด มีดังนี้

กฎข้อที่ 1 ทรานซิชันที่อยู่ในแบบจำลองเพทริเน็ตแทนการทำงาน (Operation) ของบีเมทอด

กฎข้อที่ 2 สถานะที่อยู่ในแบบจำลองเพทริเน็ตและเป็นสมาชิกของการก่อนการเปลี่ยนสถานะจะถูกแปลงเป็นเงื่อนไขของแต่ละการทำงานของบีเมทอด

กฎข้อที่ 3 กิจกรรมหลักเกิดการเปลี่ยนสถานะจะถูกแปลงให้เป็นกิจกรรมของการทำงานเมื่อเงื่อนไขเป็นจริง

เมื่อผู้วิจัยแปลงจากเพทริเน็ตไปเป็นภาษาบีเมทอดได้เรียบร้อย งานวิจัยนี้ได้ทดลองแปลงจากภาษาบีเมทอดไปเป็นภาษาอีเวนท์บีแต่ระบบที่ใช้ในการทดลองเป็นระบบที่มีพฤติกรรมการทำงานแบบตามลำดับ ซึ่งไม่ตรงกับคุณสมบัติหลักของอีเวนท์บีคือสนับสนุนการทำงานของระบบที่ทำงานพร้อมกัน งานวิจัยนี้มีประโยชน์สำหรับเริ่มต้นศึกษาเกี่ยวกับโครงสร้างทางไวยากรณ์ที่เกี่ยวข้องกับอีเวนท์บี เนื่องจากอีเวนท์บีมีโครงสร้างคล้ายกับบีเมทอดและผู้วิจัยได้ศึกษาเกี่ยวกับการแทนที่โดยรวมในความหมายของทฤษฎีบททางคณิตศาสตร์

2.2.4 งานวิจัยชื่อ From TiMo to Event-B Event-Driven Timed Mobility โดย Gabriel Ciobanu Thai Son Hoang Alin Stefanescu ปี ค.ศ. 2014

งานวิจัยนี้ [21] ได้นำเสนอเวลาที่ผ่านไปสำหรับการเคลื่อนย้ายข้อมูล (Timed Mobility) หรือมีชื่อย่อว่าทิโม (TiMo) เพื่อคำนวณกระบวนการที่มีข้อจำกัดของเวลาเข้ามาควบคลุมการเคลื่อนย้ายข้อมูลของกระบวนการจากสถานะหนึ่งไปยังสถานะหนึ่ง โดยสนใจเวลาท้องถิ่นที่มีความสามารถที่จะระบุการกระจาย

ข้อมูลของระบบแม่นยำมากขึ้น งานวิจัยนี้ออกแบบกฎทีโมเพื่อจำลองกระบวนการของระบบที่สนใจและได้ถอดความหมายของกฎออกมาให้อยู่ในรูปแบบของอีเวนท์บีสำหรับการทวนสอบความถูกต้องของการออกแบบกฎทีโมแต่อีเวนท์บีมีข้อจำกัดแนวคิดของการจับเวลาในการรีเซตเวลา งานวิจัยนี้ได้เสนอแนวคิดเรื่องตัวจับเวลาภายในให้สามารถตั้งค่าเริ่มต้นและใช้เครื่องมือโดดินเพื่อตรวจสอบความถูกต้องและวิเคราะห์คุณสมบัติต่างๆ ของระบบ

งานวิจัยดังกล่าวมีความสนใจเวลาที่ใช้สำหรับการเคลื่อนย้ายข้อมูลซึ่งมีความแตกต่างกับงานผู้วิจัย แต่งานวิจัยนี้มีประโยชน์สำหรับการศึกษาแนวทางการแปลงเพทริเน็ตกับเวลาไปเป็นอีเวนท์บีสำหรับรูปแบบข้อจำกัดของอีเวนท์บีและแนวคิดของการตั้งเวลาใหม่เมื่อจบสถานะเพื่อให้ได้ผลลัพธ์ที่สมบูรณ์แบบและไม่เกิดข้อผิดพลาดในการดำเนินงานวิจัย

2.2.5 งานวิจัยชื่อ Modeling Critical Systems with Timing Constraints in Event-B โดย Siavashi Faezeh, Wald Marina และ Tsiopoulos Leonidas ปี ค.ศ. 2013

งานวิจัยนี้ [22] ได้นำเสนอนำเสนอการการจับเวลา (Timing) ซึ่งเป็นข้อจำกัดของอีเวนท์บี ผู้วิจัยขยายขอบเขตเวลาของโมเดลเช็คกิงที่เพิ่มเวลา ออโตมาตาที่กำหนดเวลา (Timed Automata) ที่ชื่อว่า อัพพาว (UPPAAL Timed Automata) ผลงานหลักของงานวิจัยนี้คือใช้รูปแบบและการปรับแต่งคุณสมบัติการจับเวลาของอัพพาวเพื่อแปลงไปเป็นอีเวนท์บีผู้วิจัยมุ่งเน้นไปที่การทำงานร่วมกันและข้อจำกัดของเวลาทั่วไปและปรับแต่งแบบจำลองถูกแปลงออกมาเพื่อตรวจสอบคุณสมบัติของระบบได้นำเสนอรูปแบบของคุณสมบัติดังนี้

1) คุณสมบัติการดีเลย์ (Delay Properties) ในระดับนามธรรมจะถูกกำหนดจำนวนเต็มบวกซึ่งจะเพิ่มทีละหนึ่งวงจรนาฬิกาในแต่ละรอบของระบบ นาฬิกาที่ถูกกลั่นจะถูกจำลองด้วยนาฬิกาของระบบที่จะนับตามจำนวนทิค (Tick) ภายในวงจรนาฬิกาในระดับนามธรรม

2) คุณสมบัติขีดจำกัดของเวลา (Deadline) หรือเวลาที่ทับซ้อนกัน เมื่อเหตุการณ์ที่เกิดได้รับการกลั่นจะถูกกำหนดขีดจำกัดของเวลาเพื่อมั่นใจเมื่อมีเหตุการณ์เกิดขึ้น ค่าจะได้รับการป้องกันจากเหตุการณ์ที่เกี่ยวข้อง

3. งานวิจัยนี้เป็นการเสนอวิธีการในระดับแนวความคิดของข้อจำกัดในอีเวนท์บีและยังไม่ได้รับการตรวจสอบความถูกต้อง งานวิจัยดังกล่าวมีความแตกต่างกับงานของผู้วิจัย โดยงานวิจัยดังกล่าวเป็นงานวิจัยเกี่ยวกับข้อจำกัดเรื่องเวลาของอีเวนท์บี ซึ่งงานของผู้วิจัยเป็นงานวิจัยที่ศึกษาการแปลงไทมด์เพทริเน็ตเป็นอีเวนท์บี งานวิจัยดังกล่าวเป็นประโยชน์สำหรับการศึกษาข้อจำกัดเรื่องเวลาของอีเวนท์บี เพื่อให้งานของผู้วิจัยได้ผลลัพธ์ที่สมบูรณ์แบบและไม่เกิดข้อผิดพลาดในการดำเนินงานวิจัย

บทที่ 3
การแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี

วิธีดำเนินการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี ผู้วิจัยเริ่มต้นจากการสกัดและตรวจสอบส่วนประกอบของแบบจำลองไทมด์เพทริเน็ตให้อยู่ในรูปแบบของเอกซ์เอ็มแอล จากนั้นผู้วิจัยได้ออกแบบกฎการแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บี เมื่อได้กฎการแปลงไทมด์เพทริเน็ตใหเป็นอีเวนท์บีเรียบร้อยแล้ว ผู้วิจัยได้นิยามสัญลักษณ์ทางคณิตศาสตร์สำหรับแสดงในผลลัพธ์การแปลงอีเวนท์บีเพื่อให้ผลลัพธ์อีเวนท์บีโค้ดสามารถแสดงในเครื่องมือโรดินได้อย่างถูกต้อง วิธีดำเนินการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีสามารถแสดงแผนภาพกิจกรรมได้ดังรูปที่ 3.1

รูปที่ 3.1 แผนภาพกิจกรรมของวิธีดำเนินการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี

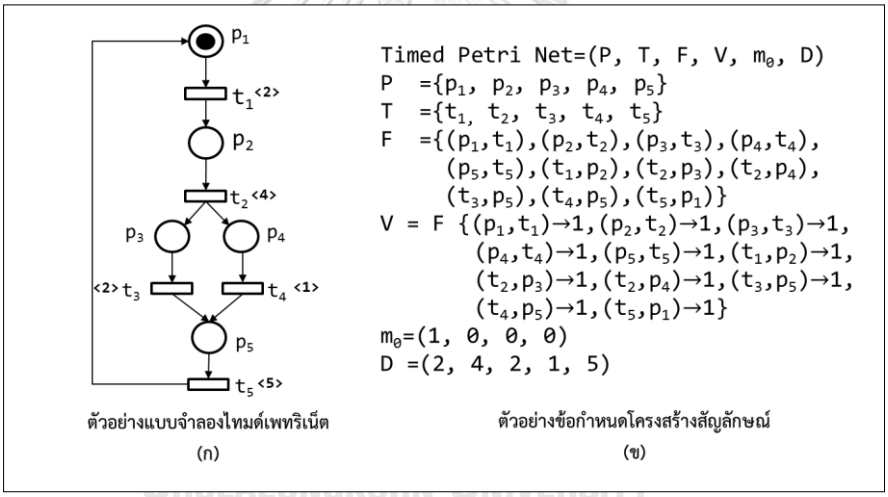
จากรูปที่ 3.1 ผู้วิจัยได้วิเคราะห์และออกแบบกฎสำหรับการตรวจสอบแบบจำลองไทมด์เพทริเน็ตสามารถอธิบายแยกเป็นสามส่วน ส่วนที่หนึ่งคือแนวคิดการแปลงแบบจำลองไทมด์เพทริเน็ตในรูปแบบแฟ้มเอกสารเอกซ์เอ็มแอล ส่วนที่สองจะกล่าวถึงการนิยามกฎการแปลงแบบจำลองไทมด์เพทริเน็ตที่อยู่ในโครงสร้างภาษาเอกซ์เอ็มแอลไปเป็นอีเวนท์บีและรูปแบบการนิยามสัญลักษณ์ในผลลัพธ์การแปลงอีเวนท์บีจะกล่าวในส่วนที่สาม

3.1 แนวคิดการแปลงแบบจำลองไทมด์เพทริเน็ตในรูปแบบแฟ้มเอกสารเอกซ์เอ็มแอล

แนวคิดสำหรับการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี เริ่มต้นจากการวิเคราะห์โครงสร้างแบบจำลองไทมด์เพทริเน็ตเป็นข้อมูลนำเข้าที่อยู่ในรูปแบบของแฟ้มเอกสารเอกซ์เอ็มแอล ผู้วิจัยได้ออกแบบแฟ้มเอกสารโครงร่างเอกซ์เอ็มแอล เพื่อทำหน้าที่ตรวจสอบความสมบูรณ์ (Validity Constraints) ของแบบจำลองไทมด์เพทริเน็ต และตรวจสอบไวยากรณ์เอกซ์เอ็มแอลที่ระบุส่วนประกอบเชิงรูปนัยของไทมด์เพทริเน็ตให้มีรูปแบบที่ถูกต้อง (Well-Formed) ตามกำหนดจาก Word Wide Web Consortium (W3C) ที่ได้กำหนดรูปแบบที่ถูกต้องของโครงสร้างเอกซ์เอ็มแอล โดยมีรายละเอียดของแบบจำลองเพทริเน็ตที่อยู่ในรูปแบบแฟ้มเอกสารเอกซ์เอ็มแอล

3.1.1 การสกัดและตรวจสอบส่วนประกอบของแบบจำลองไทมด์เพทริเน็ต

จากองค์ประกอบเชิงสัญลักษณ์ทั้งหมด 6 องค์ประกอบคือ TPN = (P, T, F, V, m_0, D) และสามารถแสดงรูปสัญลักษณ์เชิงกราฟิกได้ แสดงในรูปที่ 3.2



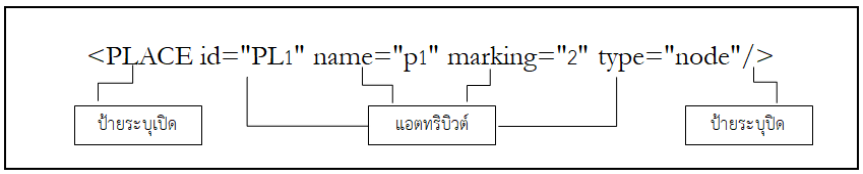
รูปที่ 3.2 ตัวอย่างแบบจำลองไทมด์เพทริเน็ต

3.1.2 การนิยามกฎการแปลงส่วนประกอบของแบบจำลองไทมด์เพทริเน็ตให้อยู่ในรูปแบบของเอกซ์เอ็มแอล

การวิเคราะห์โครงสร้างนิยามของแบบจำลองไทมด์เพทริเน็ตเพื่อนิยามให้อยู่ในรูปแบบของเอกซ์เอ็มแอลจะครอบคลุมข้อกำหนดโครงสร้างของเอกซ์เอ็มแอลที่มีลักษณะเป็นเอกสารที่มีรูปแบบถูกต้อง ซึ่งสามารถเข้าใจได้ง่ายและมีต้องมีไวยากรณ์ที่ถูกต้องสมบูรณ์ (Valid Document) ดังนั้นผู้วิจัยจึงต้องเข้าใจวิธีการใช้งานตามข้อกำหนดดังกล่าว เพื่อให้ได้แฟ้มเอกสารเอกซ์เอ็มแอลที่ระบุส่วนประกอบของไทมด์เพทริเน็ตที่มีความถูกต้องและมีไวยากรณ์ที่สมบูรณ์ โดยมีขั้นตอนดังต่อไปนี้

1) การนิยามป้ายระบุเปิดและป้ายระบุปิด

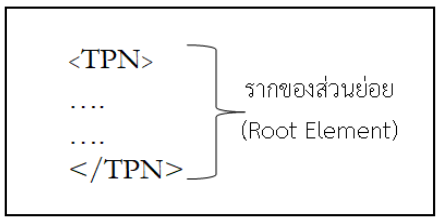
ภายในองค์ประกอบส่วนย่อยของเอกซ์เอ็มแอล ประกอบด้วยการเขียนป้ายระบุเปิด (Start Tag) และป้ายระบุปิด (End Tag) แสดงตัวอย่างดังรูปที่ 3.3



รูปที่ 3.3 รูปแบบการเขียนป้ายระบุเปิดและป้ายระบุปิด

2) การนิยามรากของส่วนย่อย

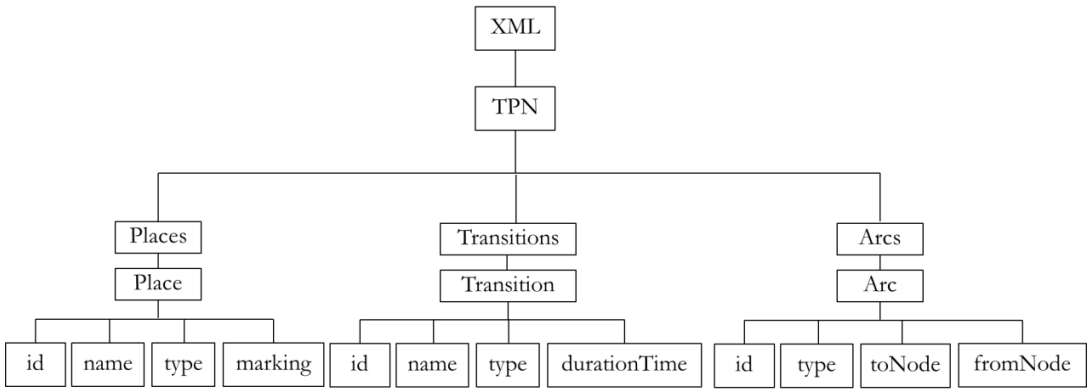
เค้าร่างเอกซ์เอ็มแอลจะต้องมีรากของส่วนย่อย (Root Element) ที่ชื่อ “TPN” เท่านั้น แสดงตัวอย่างดังรูปที่ 3.4



รูปที่ 3.4 รากของส่วนย่อย TPN ในการเขียนแบบจำลองไทมด์เพทริเน็ตในรูปแบบแฟ้มเอกซ์เอ็มแอล

3) การออกแบบเค้าร่างเอกซ์เอ็มแอลสำหรับแบบจำลองไทมด์เพทริเน็ต

จากการวิเคราะห์ส่วนประกอบของไทมด์เพทริเน็ตและทำความเข้าใจเกี่ยวกับลักษณะที่ถูกต้องของเอกซ์เอ็มแอลเรียบร้อยแล้ว ผู้วิจัยได้วิเคราะห์และเขียนแผนภาพเพื่อนิยามแบบจำลองไทมด์เพทริเน็ตสำหรับเค้าร่างของเอกซ์เอ็มแอล แสดงในรูปที่ 3.5



รูปที่ 3.5 แผนภาพออกแบบเค้าร่างไทมด์เพทริเน็ตในรูปแบบแฟ้มเอกซ์เอ็มแอล

สามารถอธิบายความหมายของเค้าร่างไทมด์เพทริเน็ตในรูปแบบแฟ้มเอกซ์เอ็มแอล ในตารางที่ 3.1

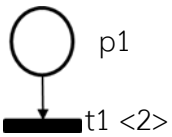
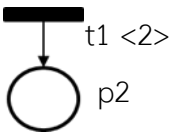
ตารางที่ 3.1 การอธิบายแผนภาพการนิยามเค้าร่างไทมด์เพทริเน็ตที่อยู่ในรูปแบบแฟ้มเอกสารเอกซ์เอ็มแอล

ส่วนประกอบ	คำอธิบาย
<TPN> ... </TPN>	รากของส่วนย่อย (Root Element) ไทมด์เพทริเน็ต
<PLACES> <PLACE id="PL_p1" name="p1" marking="2" type="node"/> </PLACES>	ส่วนประกอบของเพลส ประกอบด้วยสี่แอททริบิวต์ คือ id คือ การระบุเลขไอดีของเพลส name คือ การระบุชื่อของเพลส marking คือ การระบุจำนวนโทเค็นที่อยู่ในเพลส type คือ การระบุประเภทของแอททริบิวต์
<TRANSITIONS> <TRANSITION id="TR_t1" name="t1" durationTime ="2" type="node"/> </TRANSITIONS>	ส่วนประกอบของทรานซิชัน ประกอบด้วยสี่แอททริบิวต์ คือ id คือ การระบุเลขไอดีของเพลส name คือ การระบุชื่อของเพลส durationTime คือ การระบุระยะเวลาของทรานซิชัน type คือ การระบุประเภทของแอททริบิวต์
<ARCS> <ARC id="inputPlace" fromNode ="p1" toNode ="t1" type="node"/> <ARC> id="outputPlace" fromNode ="t1" toNode ="p2" type="node"/> </ARCS>	ส่วนประกอบของเส้นทาง ประกอบด้วยสี่แอททริบิวต์ คือ id คือ การระบุเลขไอดีของเพลส fromNode คือ การระบุอินพุตเพลส toNode คือ การระบุเอาต์พุตเพลส type คือ การระบุประเภทของแอททริบิวต์

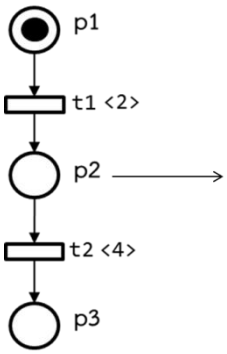
4) การนิยามเค้าร่างไทมด์เพทริเน็ตให้อยู่ในรูปแบบแฟ้มเอกซ์เอ็มแอล

จากแผนภาพการนิยามเค้าร่างไทมด์เพทริเน็ตให้อยู่ในรูปแบบแฟ้มเอกซ์เอ็มแอล ผู้วิจัยได้นำส่วนประกอบไทมด์เพทริเน็ตมาเขียนให้อยู่ในรูปแบบเค้าร่างเอกซ์เอ็มแอล แสดงตารางความสัมพันธ์ของส่วนประกอบไทมด์เพทริเน็ตและส่วนประกอบเอกซ์เอ็มแอลได้ในตารางที่ 3.2

ตารางที่ 3.2 ความสัมพันธ์ของส่วนประกอบไทมด์เพทริเน็ตและส่วนประกอบเอกซ์เอ็มแอล

ไทมด์เพทริเน็ต	ส่วนประกอบเอกซ์เอ็มแอล
	<PLACES> <PLACE id="PL_p1" **name="p1"** marking="2" type="node"/> <PLACES>
	<TRANSITIONS> <TRANSITION id="TR_t1" **name="t1"** **durationTime="3"** type="node"/> </TRANSITIONS>
	<PLACES> <PLACE id="PL_p1" name="p1" **marking="2"** type="node"/> <PLACES>
	<ARCS> <ARC id="inputPlace" **fromNode="p1"** **toNode="t1"** type="node"/> <ARCS>
	<ARCS> <ARC id="outputPlace" **fromNode="t1"** **toNode="p2"** type="node"/> <ARCS>

แสดงตัวอย่างการสกัดแต่ละส่วนประกอบของแฟ้มเอกสารไทมด์เพทริเน็ตในรูปแบบของเอกซ์เอ็มแอลที่ถูกต้องตามข้อกำหนดของเอกซ์เอ็มแอลในรูปที่ 3.6




```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<TPN>
    <PLACES>
        <PLACE id="PL_p1" name="p1" marking="1" type="node"/>
        <PLACE id="PL_p2" name="p2" marking="0" type="node"/>
        <PLACE id="PL_p3" name="p3" marking="0" type="node"/>
    <PLACES>
    <TRANSITIONS>
        <TRANSITION id="TR_t1" name="t1" durationTime="2"  type="node"/>
        <TRANSITION id="TR_t2" name="t2" durationTime="4"  type="node"/>
    </TRANSITIONS>
    <ARCS>
        <ARC id="inputPlace" fromNode="p1" toNode="t1" type="node"/>
        <ARC id="outputPlace" fromNode="t1" toNode="p2" type="node"/>
        <ARC id="inputPlace" fromNode="p2" toNode="t2" type="node"/>
        <ARC id="outputPlace" fromNode="t2" toNode="p3" type="node"/>
    </ARCS>
</TPN>
```

รูปที่ 3.6 ตัวอย่างรูปแบบแฟ้มเอกซ์เอ็มแอลขององค์ประกอบไทมด์เพทริเน็ต

3.2 การแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บี

เมื่อทำการวิเคราะห์และได้แบบจำลองไทมด์เพทริเน็ตที่อยู่ในรูปแบบแฟ้มเอกซ์เอ็มแอลที่ถูกต้องเรียบร้อยแล้ว ผู้วิจัยได้นิยามกฎการแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บี ซึ่งมีกฎการแปลงทั้งหมด 7 ข้อและนิยามสัญลักษณ์สัญลักษณ์ที่ใช้สำหรับการแปลงไทมด์เพทริเน็ตเพื่อให้ได้ผลลัพธ์การแปลงอีเวนท์บีที่ถูกต้องตามไวยากรณ์ของอีเวนท์บี

3.2.1 การนิยามกฎการแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บี

เมื่อตรวจสอบเมื่อตรวจสอบเรียบร้อยแล้วจึงนำเข้ากฎการแปลงเพื่อใช้สำหรับดำเนินการแปลงไทมด์เพทริเน็ตเป็นอีเวนท์บีโดยเขียนลงในเอกสารที่อยู่ในรูปแบบเอกเอ็มแอล สำหรับขั้นตอนแรกผู้วิจัยนำเสนอกฎการแปลงข้อที่ 1 ถึงข้อที่ 4 สำหรับการแปลงโครงสร้างของแบบจำลองไทมด์เพทริเน็ตให้อยู่ในส่วนของบริบท (Context) ในขณะที่กฎการแปลงข้อที่ 5 ถึงข้อที่ 7 คือการแปลงพฤติกรรมของไทมด์เพทริเน็ตให้อยู่ในส่วนของแมชชีน (Machine) กำหนดให้ TPN = (P, T, F, V, m_0, d) และเป้าหมายโมเดลอีเวนท์บี คือ EBM= *(*S, C, A, V, I, E, INIT*)* ดำเนินการแปลงตามกฎ ดังต่อไปนี้

กฎข้อที่ 1 กฎการแปลงส่วนประกอบเพลส

การนิยามกฎข้อที่ 1 ผู้วิจัยได้สนใจการแปลงส่วนประกอบเพลสในไทมด์เพทริเน็ต ซึ่งมีขั้นตอนการดังต่อไปนี้

1) เพิ่ม “PLACE” ในโมเดลเซต (S) คือ “PLACE” ∈ *S*
2) เพิ่มแต่ละส่วนของเพลส p ∈ *P* เป็นโมเดลค่าคงที่ (C) คือ p ∈ *C*
3) เพิ่มแต่ละส่วนของเพลส p ∈ *P* เป็นประโยคสัจพจน์ (A) คือ “partition(PLACE, {p} …)” ∈ *A*

จากการนิยามกฎการแปลงข้อที่ 1 ผู้วิจัยได้นำเสนอการแปลงเหตุการณ์เริ่มในในอีเวนท์บี ซึ่งค่าเริ่มต้นจะสามารถดูได้จากแบบจำลองไทมด์เพทริเน็ต ได้ดังตารางที่ 3.3 แสดงความสัมพันธ์ของไทมด์เพทริเน็ตและอีเวนท์บี โดยที่คอลัมน์ไทมด์เพทริเน็ตคือเพลส p1 ของแบบจำลองไทมด์เพทริเน็ต ซึ่งมีความสัมพันธ์กับส่วนประกอบบริบทในอีเวนท์บีคือ โมเดลเซตจะมี เซตของ PLACE และ เซต TRANSITION, ค่าคงที่จะคือเพลส p1 และ partition(PLACE, {p1}) คือการประกาศสัจพนธ์ของเซต PLACE

ตารางที่ 3.3 ความสัมพันธ์ไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎการแปลงส่วนประกอบเพลส

ไทมด์เพทริเน็ต	ส่วนประกอบบริบทในอีเวนท์บี
 p	**sets PLACE TRANSITION** **constants p1** **axioms** axm partition(**PLACE, {p1**})

กฎข้อที่ 2 กฎการแปลงส่วนประกอบทรานซิชัน

การนิยามกฎข้อที่ 2 ผู้วิจัยได้สนใจการแปลงส่วนประกอบทรานซิชันในไทมด์เพทริเน็ต ซึ่งมีขั้นตอนการดังต่อไปนี้

1) เพิ่ม “TRANSITION” ในโมเดลเซต (S) คือ “TRANSITION” $\in S$
2) เพิ่มแต่ละทรานซิชัน t $\in T$ ในโมเดลค่าคงที่ (C) คือ p $\in C$
3) เพิ่ม “hold”, “enabled”, “firing” และ “reset” ในโมเดลค่าคงที่ (C)
4) เพิ่ม “PREPL” และ “POSTPL” ในโมเดลค่าคงที่ (C)
5) เพิ่มแต่ละทรานซิชัน t $\in T$ ในประโยคสัจพจน์ (A) คือ “partition(TRANSITION, {t} ...)” $\in A$
6) เพิ่ม “STATUS” ในโมเดลเซต (S)
7) เพิ่ม “partition(STATUS, {hold}, {enabled}, {firing}, {reset})” ในประโยคสัจพจน์ (A)
8) เพิ่ม “POSTPL $\in$ TRANSITION $\leftrightarrow$ PLACE” ในประโยคสัจพจน์ (A)
9) สร้างเซตของอินพุตเพลส (t, p) “PREPL” เมื่อ t $\in T$, p $\in P$, preplace (t) = p, และเพิ่ม “PREPL”ในประโยคสัจพจน์ (A)
10) สร้างเซตของเอาต์พุตเพลส (t, p) “POSTPL” เมื่อ t $\in T$, p $\in P$, postplace(t) = p, และเพิ่ม “POSTPL” ในประโยคสัจพจน์ (A)

จากการนิยามกฎการแปลงข้อที่ 2 ผู้วิจัยได้นำเสนอการแปลงเหตุการณ์เริ่มในในอีเวนท์บี ซึ่งค่าเริ่มต้นจะสามารถดูได้จากแบบจำลองไทมด์เพทริเน็ต ได้ดัง

ตารางที่ 3.4 แสดงความสัมพันธ์ของไทมด์เพทริเน็ตและอีเวนท์บี โดยที่คอลัมน์ไทมด์เพทริเน็ตคือทรานซิชัน t1 ที่มีระยะเวลาเท่ากับ 3 ของแบบจำลองไทมด์เพทริเน็ต ซึ่งมีความสัมพันธ์กับส่วนประกอบบริบทในอีเวนท์บี คือ โมเดลเซตจะมี เซตของ PLACE, เซต TRANSITION และเซตสถานะ STATUS, ค่าคงที่จะคือทรานซิชัน t1, อินพุตเพลส PREPL, เอาต์พุตเพลส POSTPL และ สถานะ hold, enabled, firing และ reset การประกาศสัจพนธ์ของเซตทรานซิชัน คือ partition(TRANSITION, {t1}) สัจพนธ์ของสถานะ partition(STATUS,{hold}, {enabled}, {firing}, {reset}) และกำหนดให้อินพุตเพลสเป็นสมาชิกของเซตของทรานซิชันและเซตของเพลส PREPL $\in$ TRANSITION $\leftrightarrow$ PLACE และเอาต์พุตเพลสเป็นสมาชิกของเซตของทรานซิชันและเซตของเพลส POSTPL $\in$ TRANSITION $\leftrightarrow$ PLACE

ตารางที่ 3.4 ความสัมพันธ์ไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎการแปลงส่วนประกอบทรานซิชัน

ไทมด์เพทริเน็ต	ส่วนประกอบบริบทในอีเวนท์บี
t1 <3>	**sets PLACE TRANSITION STATUS** **constants t1 hold enabled firing reset PREPL POSTPL** **axioms** axm partition(**TRANSITION**,{t1}) axm partition(**STATUS**,{hold}, {enabled}, {firing}, {reset}) axm PREPL∈**TRANSITION ↔ PLACE** axm POSTPL∈**TRANSITION ↔ PLACE** **end**

กฎข้อที่ 3 กฎการแปลงส่วนประกอบของการไหลของเส้นทางการย้ายสถานะ

การนิยามกฎข้อที่ 3 ผู้วิจัยสนใจการแปลงส่วนประกอบของการไหลของเส้นทางการย้ายสถานะ ซึ่งมีขั้นตอนการดังต่อไปนี้

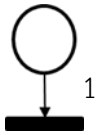
1) เพิ่ม “WEIGHT” ในโมเดลค่าคงที่ (C)
2) เพิ่ม “WEIGHT ∈ N”, “WEIGHT = 1” เป็นประโยคสัจพจน์ (A)

จากการนิยามกฎการแปลงข้อที่ 3 ผู้วิจัยได้นำเสนอการแปลงเหตุการณ์เริ่มในในอีเวนท์บี ซึ่งค่าเริ่มต้นจะสามารถดูได้จากแบบจำลองไทมด์เพทริเน็ต ได้ดัง

ตารางที่ 3.5 แสดงความสัมพันธ์ของไทมด์เพทริเน็ตและอีเวนท์บี โดยที่คอลัมน์ไทมด์เพทริเน็ต คือ ค่าน้ำหนักของการย้ายสถานะระหว่างเพลสและทรานซิชันมีค่าเท่ากับ 1 ของแบบจำลองไทมด์เพทริเน็ต ซึ่งมีความสัมพันธ์กับส่วนประกอบบริบทในอีเวนท์บี คือ ประกาศค่าคงที่ WEIGHT ที่มีสัจพนธ์ WEIGHT เป็นจำนวนเต็มที่มีค่าเท่ากับ 1

ตารางที่ 3.5 ความสัมพันธ์ระหว่างไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎการแปลงส่วนประกอบของการไหลของเส้นทางการย้ายสถานะ

ไทมด์เพทริเน็ต	ส่วนประกอบบริบทในอีเวนท์บี

	**sets PLACE TRANSITION STATUS**
	constants WEIGHT
	axioms
	  axm WEIGHT∈N
	  axm WEIGHT=1
	end

กฎข้อที่ 4 กฎการแปลงช่วงเวลาของแต่ละทรานซิชัน

การนิยามกฎข้อที่ 4 ผู้วิจัยสนใจการแปลงช่วงเวลาของแต่ละทรานซิชัน ซึ่งมีขั้นตอนการดังต่อไปนี้

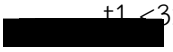
1) เพิ่ม “durationTime” ในโมเดลค่าคงที่ (C)

2) สร้างเซตของระยะเวลา d(t) “DTIME” โดยที่สมาชิกของเซต “DTIME” คือ (t, N) เมื่อ t ∈ *T* และ N คือจำนวนจริงบวกที่มีค่ามากกว่าหรือเท่ากับศูนย์ของแต่ละทรานซิชันและเพิ่ม “DTIME” ในประโยคสัจพจน์ (A)

จากการนิยามกฎการแปลงข้อที่ 4 ผู้วิจัยได้นำเสนอความสัมพันธ์ระหว่างไทมด์เพทริเน็ตไปเป็นอีเวนท์บี ได้ดังตารางที่ 3.6 แสดงความสัมพันธ์ของไทมด์เพทริเน็ตและอีเวนท์บี โดยที่คอลัมน์ไทมด์เพทริเน็ต คือ ระยะเวลาของทรานซิชัน t1 มีค่าเท่ากับ 3 ของแบบจำลองไทมด์เพทริเน็ต ซึ่งมีความสัมพันธ์กับส่วนประกอบบริบทในอีเวนท์บี คือ ประกาศค่าคงที่ t1 และ durationTime ที่มีสัจพนธ์ durationTime เป็นสมาชิกของ TRANSITION → N ที่มีค่าเท่ากับ 3

ตารางที่ 3.6 ความสัมพันธ์ไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎการแปลงช่วงเวลาของแต่ละทรานซิชัน

ไทมด์เพทริเน็ต	ส่วนประกอบบริบทในอีเวนท์บี

	**sets PLACE TRANSITION STATUS**
	constants t1 durationTime
	axioms
	  axm **durationTime** ∈ **TRANSITION** → N
	  axm **durationTime** = { t1 |->3}
	end

กฎข้อที่ 5 กฎการนิยามตัวแปร (Variables) และตัวยืนยง (Invariants) ในอีเวนท์บี

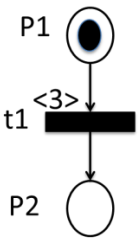
การนิยามกฎข้อที่ 5 ผู้วิจัยสนใจการนิยามตัวแปรและตัวยืนยงในอีเวนท์บี ซึ่งมีขั้นตอนการดังต่อไปนี้

1) เพิ่ม “s_PLACE”, “s_TRANSITION”, “s_MARKING”, “STAT”, “TIK”, และ “postPlace” ในโมเดลตัวแปร (V) เพื่อให้สามารถควบคุมพฤติกรรมของเน็ต

2) เพิ่ม “s_PLACE ⊆ PLACE”, “s_TRANSITION ⊆ TRANSITION”, “s_MARKING ∈ PLACE → Z”, “STAT ∈ TRANSITION → STATUS”, “TIK ∈ s_TRANSITION → N”, “postPlace ⊆ s_PLACE” และ “prePlace ⊆ s_PLACE” ในโมเดลตัวยืนยง (I) ซึ่งตัวแปรทั้งหมดในตัวยืนยงต้องมาจากตัวแปร (V) เท่านั้น

จากการนิยามกฎการแปลงข้อที่ 5 ผู้วิจัยได้นำเสนอการนิยามตัวแปรและตัวยืนยงในอีเวนท์บี ซึ่งจะต้องมีความสอดคล้องกับโครงสร้างของแบบจำลองไทมด์เพทริเน็ต ได้ดังตารางที่ 3.7 แสดงความสัมพันธ์ของไทมด์เพทริเน็ตและอีเวนท์บี โดยที่คอลัมน์ไทมด์เพทริเน็ต คือ ของแบบจำลองไทมด์เพทริเน็ตที่ประกอบด้วยเพลส p1 และ p2, ทรานซิชัน t1 ที่มีระยะเวลาเท่ากับ 3 ซึ่งมีความสัมพันธ์กับส่วนประกอบบริบทในอีเวนท์บี คือ มีการประกาศตัวแปร s_PLACE จะสอดคล้องกับเพลสในไทมด์เพทริเน็ต, s_TRANSITION จะสอดคล้องกับทรานซิชันในไทมด์เพทริเน็ต, s_MARKING จะสอดคล้องกับมาก์กิงในไทมด์เพทริเน็ต, STAT ประกาศเพื่อให้ทราบสถานะของแต่ละทรานซิชัน, TIK คือตัวนับเวลาในไทมด์เพทริเน็ต, postPlace คือตัวเก็บเพลสของอินพุตเพลสของการยิงในแต่ละทรานซิชัน และ preplace คือ ตัวเก็บเพลสของเอาต์พุตเพลสของการยิงในแต่ละทรานซิชัน

ตารางที่ 3.7 ความสัมพันธ์ไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎกานิยามตัวแปรและตัวยืนยงในอีเวนท์บี

ไทมด์เพทริเน็ต	ส่วนประกอบแมชชีนในอีเวนท์บี
	**variables s_PLACE s_TRANSITION s_MARKING STAT TIK postPlace prePlace** **invariants** **inv1 s_PLACE ⊆ PLACE** **inv2 s_TRANSITION ⊆ TRANSITION** **inv3 s_MARKING ∈ TRANSITION → Z** **inv4 STAT ∈ TRANSITION → STATUS** **inv5 TIK ∈ s_TRANSITION → N** **inv6 postPlace ⊆ s_PLACE** **inv7 prePlace ⊆ s_PLACE** **end**

กฎข้อที่ 6 กฎการแปลงเหตุการณ์เริ่มต้น (Initialisation Events)

การนิยามกฎข้อที่ 6 ผู้วิจัยสนใจการแปลงเหตุการณ์เริ่มต้นในอีเวนท์บี ซึ่งมีขั้นตอนการดังต่อไปนี้

1) เพิ่มประโยค “s_PLACE := PLACE” and “s_TRANSITION := TRANSITION” ในเหตุการณ์เริ่มต้น (INIT)

2) สร้างเซตของมาร์กกิง (p,q) “s_MARKING” เมื่อ p ∈ *P*, q = $m_0(p)$ และ “s_MARKING” คือประโยคในเหตุการณ์เริ่มต้น (INIT)

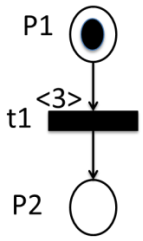
3) สร้างเซตสถานะ (t,u) “STAT” เมื่อ t ∈ T, u ∈ {hold} และเพิ่ม “STAT” ในเหตุการณ์เริ่มต้น (INIT)

4) สร้างเซตทิค (t,0) “TIK” เมื่อ t ∈ T และเพิ่ม “TIK” ในเหตุการณ์เริ่มต้น (INIT)

5) เพิ่มประโยค “postPlace: = Ø” และ “preplace := Ø” ในเหตุการณ์เริ่มต้น (INIT) เพื่อให้ “postPlace” และ “prePlace” สามารถรีเซตให้เป็นเซตว่าง

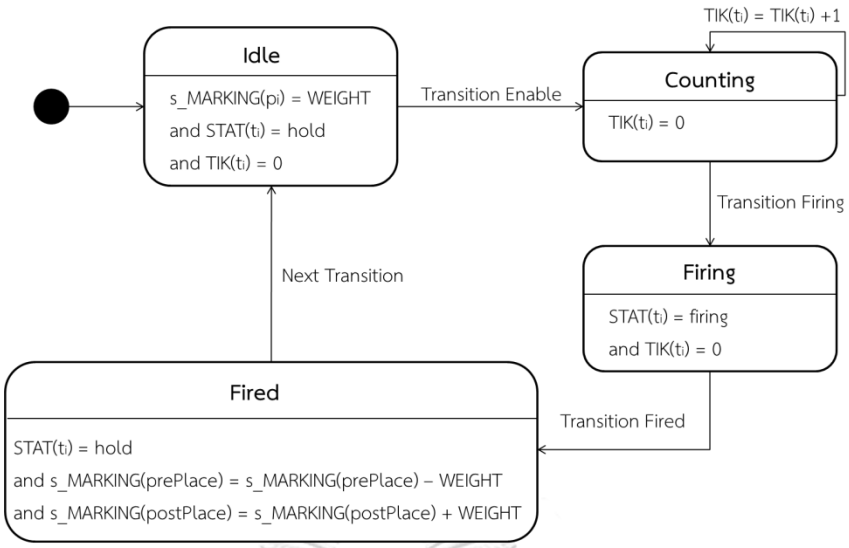
จากการนิยามกฎการแปลงข้อที่ 6 ผู้วิจัยได้นำเสนอการแปลงเหตุการณ์เริ่มในในอีเวนท์บี ซึ่งค่าเริ่มต้นจะสามารถดูได้จากแบบจำลองไทมด์เพทริเน็ต แสดงได้ดังตารางที่ 3.8 แสดงความสัมพันธ์ของไทมด์เพทริเน็ตและอีเวนท์บี โดยที่คอลัมน์ไทมด์เพทริเน็ต คือ ของแบบจำลองไทมด์เพทริเน็ตที่ประกอบด้วย เพลส p1 และ p2, ทรานซิชัน t1 ที่มีระยะเวลาเท่ากับ 3 ซึ่งมีความสัมพันธ์กับส่วนประกอบบริบทในอีเวนท์บี คือ ตัวแปะ s_PLACE มีค่าเท่ากับเซต PLACE, s_TRANSITION มีค่าเท่ากับเซต TRANSITION, s_MARKING มีค่าเท่ากับ เพลส p1 มีจำนวนโทเค็นเท่ากับ 1, เพลส p2 มีจำนวนโทเค็นเท่ากับ 0, ตัวนับเวลาของทรานซิชัน t1 มีค่าเริ่มต้นเท่ากับ 0, เซตของเพลสของอินพุตเพลสและเอาต์พุตเพลสมีค่าเท่ากับเซตว่าง

ตารางที่ 3.8 ความสัมพันธ์ไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยกฎการแปลงเหตุการณ์เริ่มต้น

ไทมด์เพทริเน็ต	ส่วนประกอบแมชชีนในอีเวนท์บี
	**events** **event Initialisation** **then** act1 **s_PLACE:= PLACE** act2 **s_TRANSITION:= TRANSITION** act3 **s_MARKING:= {p1 |-> 1, p2 |-> 0}** act4 **STAT:= {t1 |-> hold}** act5 **TIK:= {t1 |->0}** act6 **postPlace:= Ø** act7 **prePlace:= Ø** **end**

กฎข้อที่ 7 กฎการแปลงเหตุการณ์แบบไดนามิก (Dynamic Events)

ผู้วิจัยได้ทำการวิเคราะห์พฤติกรรมของไทมด์เพทริเน็ตและสกัดออกมาเพื่อถอดแบบออกมาให้อยู่ในโมเดลของเหตุการณ์ในอีเวนท์บีจะประกอบด้วยสถานะ Idle เมื่อเพลสมีจำนวนโทเค็นเท่ากับ WEIGHT สถานะของทรานซิชัน STAT(t_i) เท่ากับ “hold” และตัวนับเวลา TIK(t_i) เท่ากับ 0 จะเกิดเหตุการณ์ Transition Enable แล้วจะอยู่ในสถานะ Counting เมื่อตัวนับเวลา TIK(t_i) ของเท่าซิชันมีค่าเท่ากับระยะเวลาของทรานซิชัน durationTime(t_i) แล้วจะเกิดเหตุการณ์ Transition Firing แล้วเข้าสู่สถานะ Firing เมื่อ สถานะของทรานซิชัน STAT(t_i) เท่ากับ “firing” และ ตัวนับเวลา TIK(t_i) เท่ากับ 0 จะเกิดเหตุการณ์ Transition Fired และเข้าสู่สถานะ Fired เมื่อสถานะของทรานซิชัน STAT(t_i) เท่ากับ “hold” และลบโทเค็นในอินพุตเพลส s_MARKING(prePlace) ออกหนึ่งโทเค็นหนึ่งโทเค็นและเพิ่มโทเค็นในเอาต์พุตเพลส s_MARKING(postPlace) เข้าไปหนึ่งโทเค็น จากนั้นจะเกิดทรานซิชันถัดไป แสดงลำดับการเกิดเหตุการณ์ได้ดังรูปที่ 3.7



รูปที่ 3.7 แผนภาพการทำงานของเหตุการณ์แบบไดนามิกภายในอีเวนท์บีโค้ด

1) เหตุการณ์ Enabled

ลักษณะการทำงานของการเปิดใช้งานของแต่ละทรานซิชัน *tt* จะอธิบายด้วยเหตุการณ์ “Enabled” ซึ่งมีสองเงื่อนไข ซึ่งเงื่อนไขที่หนึ่ง (grd1) คือการตรวจสอบให้ว่าทรานซิชัน *t* ใดๆ เป็นสมาชิกของ s_TRANSITION และ STAT(t) อยู่ในสถานะ “hold” เงื่อนไขที่สอง (grd2) คือการตรวจสอบว่าทุกๆ เพลส pp เป็นสมาชิกของ PREPL[{t}] แล้ว PREPL[{*t*}] เป็นสับเซตของ s_PLACE และจำนวนมาร์กกิงของเพลส s_MARKING(*p*) มีค่ามากกว่าหรือเท่ากับ WEIGHT เมื่อใดก็ตามที่เงื่อนไขที่หนึ่งและเงื่อนไขที่สอง (grd1 และ grd2) เป็นจริงแล้ว STAT(t) จะเปลี่ยนเป็น “enabled” แสดงเหตุการณ์ “Enabled” และเพิ่มเพลสของ PREPL[{t}] ในเซตของ prePlace สามารถแสดงเหตุการณ์ Enabled สำหรับอีเวนท์บีโค้ด ดังรูปที่ 3.8

event Enabled
any *t*
where
grd1 (*t* ∈ s_TRANSITION) ∧ (STAT(*t*) = hold)
grd2 ∀*pp*·(*pp*∈ PREPL[{*t*}] ⇒ PREPL[{*t*}]⊆ s_PLACE ∧ (s_MARKING(*pp*)=WEIGHT))
then
act1 STAT(t) ≔ enabled
act2 prePlace ≔ PREPL[{t}]

รูปที่ 3.8 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ Enabled สำหรับอีเวนท์บีโค้ด

2) เหตุการณ์ Counting

ตัวนับเวลาถูกอธิบายโดยมีเหตุการณ์ "Counting" ต่อไปนี้ ในทำนองเดียวกัน เงื่อนไขมีการกำหนดด้วยสองเงื่อนไข ซึ่งเงื่อนไขที่หนึ่ง (grd1) คือการตรวจสอบให้ว่าทรานซิชัน *t* ใดๆ เป็นสมาชิกของ s_TRANSITION และ STAT (t) อยู่ในสถานะ “enabled” เงื่อนไขที่สอง (grd2) คือการตรวจสอบว่าทิคของทรานซิชันใดๆ TIK(t) น้อยกว่าระยะเวลา d(t) เมื่อใดก็ตามที่ทั้งสองเงื่อนไข (grd1 และ grd2) เป็นจริงแล้ว TIK(t) จะถูกนับเพิ่มขึ้นทีละหนึ่ง สามารถแสดงเหตุการณ์ Counting สำหรับอีเวนท์บีโค้ด ดังรูปที่ 3.9

```
event Counting
 any t
 where
  grd1 (t ∈ s_TRANSITION) ∧ (STAT(t) = enabled)
  grd2 (TIK(t) < durationTime(t))
 then
  act1 TIK(t) ≔ TIK(t) +1
 end
```

รูปที่ 3.9 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ Counting สำหรับอีเวนท์บีโค้ด

3) เหตุการณ์ Fire

ลักษณะการทำงานของการยิงของแต่ละทรานซิชัน t จะอธิบายด้วยเหตุการณ์ “Fire” ซึ่งมีสองเงื่อนไข ซึ่งเงื่อนไขที่หนึ่ง (grd1) คือการตรวจสอบให้ว่าทรานซิชัน *t* ใดๆ เป็นสมาชิกของ s_TRANSITION, TIK (t) มีค่าเท่ากับระยะเวลา durationTime(t) และ STAT(t) อยู่ในสถานะ “enabled” เงื่อนไขที่สอง (grd2) เป็นการตรวจสอบว่าเพลส p ใดๆ เป็นสมาชิกของ s_PLACE, p เป็นสมาชิกของ PREPL [{t}] และ POSTPL[{t}] เป็นสับเซตของ s_PLACE เมื่อใดก็ตามที่ทั้งสองเงื่อนไข (grd1 และ grd2) เป็นจริงแล้ว STAT (t) จะถูกเปลี่ยนเป็น “firing” และ POSTPL ของทรานซิชัน t จะถูกเพิ่มเข้าไปในเซต postPlace สามารถแสดงเหตุการณ์ Fire สำหรับอีเวนท์บีโค้ด ดังรูปที่ 3.10

event Fire
any *t p*
where
grd1 (*t* ∈ s_TRANSITION) ∧ (TIK(*t*) = durationTime(*t*)) ∧ (STAT(*t*) = enabled)
grd2 (*p* ∈ s_PLACE) ∧ (*p* ∈ PREPL[{*t*}]) ∧ (POSTPL[{*t*}] ⊆ s_PLACE)
then
act1 STAT(*t*) ≔ firing
act2 postPlace ≔ postPlace ∪ POSTPL[{*t*}]
end

รูปที่ 3.10 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ Fire สำหรับอีเวนท์บีโค้ด

4) เหตุการณ์ resetIndividualTransition

พฤติกรรมของตัวนับเวลาและสถานะของทรานซิชัน t จะถูกรีเซต ตามที่อธิบายไว้ในเหตุการณ์ "resetIndividualTransition" ซึ่งมีสองเงื่อนไข ซึ่งเงื่อนไขที่หนึ่ง (grd1) คือการตรวจสอบว่าทรานซิชัน t ใดๆ เป็นสมาชิกของ dom(TIK), STAT (t) คือ “firing”, เซต postPlace เป็นเซตว่างและ POSTPL [{t}] เป็นเซตย่อยของ s_PLACE เงื่อนไขที่สอง (grd2) คือการตรวจสอบว่าเพลส p ใดๆ เป็นสมาชิกของ s_PLACE และเพลส p เป็นสมาชิกของ PREPL [{t}] เมื่อใดก็ตามที่ทั้งสองเงื่อนไข (grd1 และ grd2) เป็นจริงแล้ว STAT(t) ถูกรีเซตเป็น “hold”และ TIK(t) จะถูกรีเซตเป็น 0 สามารถแสดงเหตุการณ์ Enabled สำหรับอีเวนท์บีโค้ด ดังรูปที่ 3.11

event resetIndividualTransition
any *t p*
where
grd1 (*t* ∈ dom(TIK)) ∧ (STAT(*t*)=reset) ∧ (postPlace=∅) ∧ (POSTPL[{*t*}] ⊆ s_PLACE)
grd2 (*p* ∈ s_PLACE) ∧ (*p* ∈ PREPL[{*t*}])
then
act1 STAT(*t*) ≔ hold
act2 TIK(*t*) ≔ 0

รูปที่ 3.11 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ resetIndividualTransition สำหรับอีเวนท์บีโค้ด

5) เหตุการณ์ resetConcurrentTransition

พฤติกรรมของตัวนับเวลาใด ๆ และสถานะของทรานซิชัน *t* ใดๆ จะถูกรีเซตตามที่อธิบายไว้ในเหตุการณ์ "resetConcurrentTransition" ซึ่งมีสองเงื่อนไข เงื่อนไขที่หนึ่ง (grd1) คือการตรวจสอบว่าทรานซิชัน *t* ใดๆ เป็นสมาชิกของ s_TRANSITION นอกจากนี้ STAT (t) อยู่ในสถานะ “enabled”, TIK (t) ไม่เท่ากับระยะเวลาของทรานซิชัน duratioTime(t) และเพลส p ใดๆ คือสมาชิกของ PREPL [{t}] เงื่อนไขที่สอง (grd2) คือการตรวจสอบว่าเพลส p ใดๆ เป็นสมาชิกของ s_PLACE และ มาร์กกิงของเพลส s_MARKING (p) มีค่าเป็นศูนย์ เมื่อใดก็ตามที่ทั้งสองเงื่อนไข (grd1 และ grd2) เป็นจริงแล้ว STAT (t) ถูกตั้งค่าเป็น “reset” และ TIK (t) จะถูกรีเซตเป็น สามารถแสดงเหตุการณ์ resetConcurrentTransition สำหรับอีเวนท์บีโค้ด ดังรูปที่ 3.12

event resetConcurrentTransition
  any *t*
  where
    grd1 $\forall pp \cdot (pp \in \text{PREPL}[\{t\}] \Rightarrow \text{PREPL}[\{t\}] \subseteq \text{s\_PLACE} \wedge (\text{s\_MARKING}(pp)=0))$
    grd2 STAT(*t*)=firing
  then
    act1 STAT(*t*) ≔ reset
end

รูปที่ 3.12 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ resetConcurrentTransition สำหรับอีเวนท์บีโค้ด

6) เหตุการณ์ removeToken

โทเค็นของตำแหน่งอินพุตเพลสของบางทรานซิชัน *t* จะถูกเอาออกตามที่อธิบายไว้ในเหตุการณ์ "removeToken" ซึ่งมีสองเงื่อนไข เงื่อนไขที่หนึ่ง (grd1) คือการตรวจสอบอินพุตเพลส pln เป็นสมาชิกของ PREPL [{t}], POSTPL [{t}] เป็นเซตย่อยของ s_PLACE และเซต postPlace เป็นเซตว่าง เงื่อนไขที่สอง (grd2) คือการตรวจสอบอินพุตเพลส pln เป็นสมาชิกของ s_PLACE, จำนวนโทเค็นของ s_MARKING (pln) มากกว่าศูนย์และ STAT(t) อยู่ในสถานะ “reset” เมื่อใดก็ตามที่ทั้งสองเงื่อนไข (grd1 และ grd2) เป็นจริง จำนวนของโทเค็นใน s_MARKING(pln) ลดลงโดยการลดลงตามจำนวนของ WEIGHT และ STAT(t) อยู่ในสถานะ “hold” สามารถแสดงเหตุการณ์ removeToken สำหรับอีเวนท์บีโค้ด ดังรูปที่ 3.13

event removeToken
  any *t pln*
  where
    grd1 (*pln* ∈dom(s_MARKING)) ∧(*pln* ∈PREPL[{*t*}]) ∧
      (POSTPL[{*t*}] ⊆ s_PLACE) ∧ (s_MARKING(*pln*)≠0)
    grd2 ∀*pp*·(*pp*∈ prePlace⇒prePlace ⊆s_PLACE ∧ (s_MARKING(*pp*)≥WEIGHT))
    grd3 STAT(*t*)=firing
  then
    act1 s_MARKING(*pln*) ≔ s_MARKING(*pln*) − WEIGHT
end

รูปที่ 3.13 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ removeToken สำหรับอีเวนท์บีโค้ด

7) เหตุการณ์ e_DynamicTransition

โทเค็นของเอาต์พุตเพลสของทรานซิชัน t ใดๆ ที่อยู่ในสถานะ “การยิง” จะถูกเพิ่มตามที่อธิบายไว้ในเหตุการณ์ "e_DynamicTransition" ซึ่งมีสามเงื่อนไข เงื่อนไขที่หนึ่ง (grd1) คือการตรวจสอบทรานซิชัน t ใดๆ เป็นสมาชิกของ s_TRANSITION ขณะที่เอาต์พุตเพลส pOut เป็นสมาชิกของ s_PLACE และ STAT(t) อยู่ในสถานะ “enabled” เงื่อนไขที่สอง (grd2) คือการตรวจสอบเซต postPlace ไม่ใช่เซตว่างและ pOut เป็นสมาชิกของ postPlace และจำนวนโทเค็นใน s_MARKING(pOut) มีค่าเท่ากับ 0 เงื่อนไขที่สาม (grd3) คือการตรวจสอบอินพุตเพลส pln เป็นสมาชิกของ s_PLACE และ pln เป็นสมาชิกของ PREPL [{t}] เมื่อใดก็ตามที่ทั้งสามเงื่อนไข (grd1, grd2 และ grd3) เป็นจริงจำนวนของโทเค็นใน s_MARKING(pln) จะเพิ่มขึ้นตามจำนวนของ WEIGHT และ POSTPL ของทรานซิชัน t จะถูกลบออกจากเซต PostPlace สามารถแสดงเหตุการณ์ e_DynamicTransition สำหรับอีเวนท์บีโค้ดดังรูปที่ 3.14

event e_DynamicTransition

any *t pIn pOut*

where

grd1 (*t*∈ s_TRANSITION) ∧(*pOut*∈ s_PLACE) ∧(STAT(*t*) = firing) ∧(*pOut*∈ POSTPL[{*t*}])

grd2 (postPlace ≠ ∅) ∧(*pOut* ∈ postPlace) ∧s_MARKING(pOut)=0

grd3 (*pIn*∈ s_PLACE) ∧(*pIn*∈ PREPL[{*t*}])

then

act1 s_MARKING(*pOut*) ≔ s_MARKING(*pOut*) + WEIGHT

act2 postPlace ≔ postPlace ∖ {*pOut*}

end

รูปที่ 3.14 การแปลงพฤติกรรมของไทมด์เพทริเน็ตเพื่อให้ได้เหตุการณ์ e_DynamicTransition สำหรับอีเวนท์บีโค้ด

จากการนิยามกฎการแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บีด้านบน ผู้วิจัยดำเนินการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีโดยเขียนลงในเอกสารที่อยู่ในรูปแบบเอกซ์เอ็มแอลได้ โดยแสดงตัวอย่างแบบง่ายของกฎการแปลงในรูปที่ 3.15

ส่วนประกอบของไทมด์เพทริเน็ตในรูปแบบโครงสร้างเอกซ์เอ็มแอล	ส่วนประกอบของอีเวนท์บี
<Places> <place id="PL_1" name="p1" initialMarking="1" type="node"/> <place id="PL_2" name="p2" initialMarking="1" type="node"/> </Places>	**sets** PLACE **constants** p1 p2 **axioms** // set of Place @axm1 partition(PLACE,{p1,p2}) … **end**

<Transitions>
<Transition id="TR_1" name="t1" durationTime="2" type="node"/>
<Transition id="TR_2" name="t2" durationTime="4" type="node"/>
</Transitions>
	sets TRANSITION
	constants t1 t2 durationTime
	axioms
	// set of transition
	@axm2 partition(TRANSITION,{t1,t2})
	// duration time n; n is a natural number
	@axm3 durationTime∈ TRANSITION → ℕ
	@axm4 durationTime={t1 ↦2, t1 ↦4}
	…
	End

<Arcs> <Arc id="in_1" type="inputPlace" fromNode="PL_1" toNode="TR_1"/> </Arcs>	**constants** pre **axioms** … // input-place function @amx5 pre ∈ PLACE ⇸TRANSITION @axm6 pre={p1 ↦t1} …
</Arcs> <Arc id="out_1" type="outputPlace" fromNode="TR_1" toNode="PL_2"/> … </Arcs>	**constants** post **axioms** … // output-place function @amx7 post ∈ TRANSITION ⇸PLACE @axm8 post={t1 ↦p2} …

รูปที่ 3.15 ตัวอย่างกฎการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี

3.3 รูปแบบการนิยามสัญลักษณ์ทางคณิตศาสตร์สำหรับแสดงในผลลัพธ์การแปลงอีเวนท์บี

อีเวนท์บีเป็นวิธีการเชิงรูปนัยโดยใช้วิธีการพิสูจน์โดยทฤษฎีทางคณิตศาสตร์ ซึ่งเครื่องมือโรดินเป็นเครื่องมือที่สนับสนุนการทวนสอบอีเวนท์บีและจำเป็นต้องมีความเข้าใจสัญลักษณ์ทางคณิตศาสตร์ที่ภายในเครื่องโรดิน [11] ดังนั้นผู้วิจัยจึงได้นิยามสัญลักษณ์ทางคณิตศาสตร์สำหรับผลลัพธ์การแปลงอีเวนท์บี เพื่อสามารถนำไปทวนสอบในเครื่องมือโรดินได้อย่างถูกต้อง แบ่งเป็นสัญลักษณ์สำหรับอักขระที่ไม่ใช่แอสกี้, สัญลักษณ์สำหรับแคลคูลัสบริสุทธิ์, สัญลักษณ์สำหรับความสัมพันธ์และฟังก์ชันและสัญลักษณ์สำหรับการประโยคเงื่อนไข อธิบายรายละเอียดในตารางที่ 3.9, ตารางที่ 3.10, ตารางที่ 3.11 และตารางที่ 3.12

สำหรับการพัฒนาเครื่องมือการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีโดยอัตโนมัติ ให้มีความสามารถแปลงแบบจำลองเพทริเน็ตให้เป็นอีเวนท์บีโค้ดได้ ผู้วิจัยต้องเข้าใจไวยากรณ์และความหมายของอีเวนท์บี รวมถึงการเข้าใจสัญลักษณ์ทางคณิตศาสตร์ที่ใช้ภายในอีเวนท์บี ดังนั้นผู้วิจัยจึงได้สัญลักษณ์ที่ใช้สำหรับการแปลงไทมด์เพทริเน็ตเพื่อให้ได้ผลลัพธ์การแปลงอีเวนท์บี ซึ่งจะปรากฏเป็นสัญลักษณ์ภายในเครื่องมือโรดิน

ตารางที่ 3.9 อธิบายสัญลักษณ์สำหรับอักขระที่ไม่ใช่แอสกี้กับค่าของชุดอักขระยูนิโค้ด

ชื่อสัญลักษณ์	สัญลักษณ์ในผลลัพธ์การแปลงอีเวนท์บี	สัญลักษณ์ภายในเครื่องมือโรดิน
Set of natural numbers	NAT	N
Set of positive numbers	NAT1	N1
Set of integers	INT	Z

ตารางที่ 3.10 อธิบายสัญลักษณ์สำหรับแคลคูลัสบริสุทธิ์กับค่าของชุดอักขระยูนิโค้ด

ชื่อสัญลักษณ์	สัญลักษณ์ในผลลัพธ์การแปลงอีเวนท์บี	สัญลักษณ์ภายในเครื่องมือโรดิน
Left parenthesis	(	(
Right parenthesis	)	)
Logical implication	=>	⇒
Logical and	&	∧
Logical or	\ /	∨
For all	!	∀
There exists	#	∃

ตารางที่ 3.11 อธิบายสัญลักษณ์สำหรับความสัมพันธ์และฟังก์ชันกับค่าของชุดอักขระยูนิโค้ด

ชื่อสัญลักษณ์	สัญลักษณ์ในผลลัพธ์การแปลงอีเวนท์บี	สัญลักษณ์ภายในเครื่องมือโรดิน
Maplet	|->	↦
Empty set	{}	∅
Union	\ /	∪
Total function	-->	⟶
Relation	<->	↔

ตารางที่ 3.12 อธิบายสัญลักษณ์สำหรับการประโยคเงื่อนไขกับค่าของชุดอักขระยูนิโค้ด

ชื่อสัญลักษณ์	สัญลักษณ์ในผลลัพธ์การแปลงอีเวนท์บี	สัญลักษณ์ภายในเครื่องมือโรดิน
Equals sign	=	=
Not equal to	/=	≠
Less-than sign	<	<
Less than or equal to	<=	≤
Greater-than sign	>	>
Greater than or equal to	>=	≥
Element of	:	∈
Subset of	<:	⊂

ตัวอย่างการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี

การแสดงตัวอย่างของการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี ผู้วิจัยเลือกแบบจำลองไทมด์เพทริเน็ตรูปที่ 3.6 เพื่อให้ได้แบบจำลองไทมด์เพทริเน็ตที่อยู่ในรูปแบบของเอกซ์เอ็มแอล จากนั้นผู้วิจัยเริ่มดำเนินการแปลง โดยใช้กฎการแปลงในหัวข้อที่ 3.2.1 คือ

1) การแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบบริบท ซึ่งเป็นส่วนหนึ่งของอีเวนท์บี โดยใช้กฎการแปลงข้อที่ 1 ถึง กฎการแปลงข้อที่ 4 แสดงดังรูปที่ 3.16

```
context c_example
sets PLACE TRANSITION STATUS
constants p1 p2 p3 t1 t2 hold enabled firing reset
          durationTime WEIGHT PREPL POSTPL
axioms
    @axm0 partition(PLACE,{p1},{p2},{p3})
    @axm1 partition(TRANSITION,{t1},{t2})
    @axm2 partition(STATUS,{hold},{enabled},{firing},{reset})
    @axm3 PREPL : TRANSITION<->PLACE
    @axm4 PREPL={t1|->p1,t2|->p2}
    @axm5 POSTPL : TRANSITION<->PLACE
    @axm6 POSTPL={t1|->p2,t2|->p3}
    @axm7 WEIGHT : NAT
    @axm8 WEIGHT=1
    @axm9 durationTime : TRANSITION --> NAT
    @axm10 durationTime= {t1|->2,t2|->4}
end
```

กฎข้อที่ 1 กฎข้อที่ 2 กฎข้อที่ 3 กฎข้อที่ 4

รูปที่ 3.16 ตัวอย่างการแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบบริบทโดยใช้กฎการแปลงข้อที่ 1 ถึง กฎการแปลงข้อที่ 4

2) การแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบแมชชีน ซึ่งเป็นส่วนหนึ่งของอีเวนท์บีโดยใช้กฎการแปลงข้อที่ 5 ถึง กฎการแปลงข้อที่ 7

ผู้วิจัยใช้กฎการแปลงข้อที่ 5 สำหรับการแปลงเพื่อให้ได้ส่วนประกอบแมชชีนซึ่งอยู่ในส่วนของการประกาศตัวแปร variables และ invariants กฎการแปลงข้อที่ 6 สำหรับการแปลงเพื่อให้ได้ส่วนประกอบแมชชีนซึ่งอยู่ในส่วนของเหตุการณ์ “Initialisation” แสดงตัวอย่างดังรูปที่ 3.17

```
machine m_example sees c_example
variables s_PLACE s_TRANSITION s_MARKING STAT TIK postPlace prePlace
invariants
    @inv1 s_PLACE <: PLACE
    @inv2 s_TRANSITION <: TRANSITION
    @inv3 s_MARKING : s_PLACE --> INT
    @inv4 STAT : TRANSITION --> STATUS
    @inv5 TIK : s_TRANSITION --> NAT
    @inv6 postPlace <: s_PLACE
    @inv7 prePlace <: s_PLACE
events
event INITIALISATION
then
    @act1 s_PLACE := PLACE
    @act2 s_TRANSITION := TRANSITION
    @act3 s_MARKING := {p1|->2,p2|->0,p3|->0}
    @act4 STAT := {t1|->hold,t2|->hold}
    @act5 TIK := {t1|->0,t2|->0}
    @act6 postPlace := {}
    @act7 prePlace := {}
end
```

กฎข้อที่ 5 กฎข้อที่ 6

รูปที่ 3.17 ตัวอย่างการแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบแมชชีนโดยใช้กฎการแปลงข้อที่ 5 ถึง กฎการแปลงข้อที่ 6

สำหรับการแปลงพฤติกรรมเปิดใช้งานและถูกยิงของแต่ละทรานซิชันของไทมด์เพทริเน็ต สามารถแปลงให้อยู่ในรูปแบบของเหตุการณ์ Enable และเหตุการณ์ Fired โดยใช้กฎข้อที่ 7 แสดงตัวอย่างดังรูปที่ 3.18

```
event Enabled
any  t
where
    @grd1 (t : s_TRANSITION) & (STAT(t)=hold)
    @grd2 !pp.(pp : PREPL[{t}]=>PREPL[{t}] <: s_PLACE
        & (s_MARKING(pp)=WEIGHT))
then
    @act1 STAT(t) := enabled
    @act2 prePlace := PREPL[{t}]
end
event Fired
any t p
where
    @grd1 (t : s_TRANSITION) & (TIK(t)=durationTime(t))
        & (STAT(t)=enabled)
    @grd2 (p : s_PLACE) & (p : PREPL[{t}])
        & (POSTPL[{t}] <: s_PLACE)
then
    @act1 STAT(t) := firing
    @act2 postPlace := postPlace\/POSTPL[{t}]
end
```

กฎข้อที่ 7 เหตุการณ์ Enable และ เหตุการณ์ Fired

รูปที่ 3.18 ตัวอย่างการแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบแมชชีนโดยใช้กฎการแปลงข้อที่ 7 สำหรับเหตุการณ์ Enable และ เหตุการณ์ Fired

สำหรับการแปลงพฤติกรรมเพิ่มโทเค็นให้เอาต์พุตเพลสเมื่อทรานซิชันนั้นถูกยิงแสดงในเหตุการณ์ e_DynamicTransition และการนับเวลาของแต่ละทรานซิชันแสดงในเหตุการณ์ Counting โดยใช้กฎข้อที่ 7 แสดงตัวอย่างดังรูปที่ 3.19

```
event e_DynamicTransition
    any t pIn pOut
    where
    @grd1 (t : s_TRANSITION) & (pOut : s_PLACE)
        & (STAT(t)=firing) & (pOut : POSTPL[{t}])
    @grd2 (postPlace/={}) & (pOut : postPlace)
        & s_MARKING(pOut)=0
    @grd3 (pIn : s_PLACE) & (pIn : PREPL[{t}])
then
    @act1 s_MARKING(pOut) := s_MARKING(pOut) + WEIGHT
    @act2 postPlace := postPlace\{pOut}
end
event Counting
any t
where
    @grd1 (t : s_TRANSITION) & (STAT(t)=enabled)
    @grd2 (TIK(t)<durationTime(t))
then
    @act1 TIK(t) := TIK(t) + 1
end
```

กฎข้อที่ 7 เหตุการณ์ e_DynamicTransition และ เหตุการณ์ Counting

รูปที่ 3.19 ตัวอย่างการแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบแมชชีนโดยใช้กฎการแปลงข้อที่ 7 สำหรับเหตุการณ์ e_DynamicTransition และ เหตุการณ์ Counting

เมื่อแต่ละทรานซิชันถูกยิง สถานะและตัวนับเวลาของแต่ละทรานซิชันจะต้องถูกรีเซต จะแสดงในเหตุการณ์ resetIndividualTransition และ resetConcurrentTransition เมื่อดำเนินการรีเซตสถานะและตัวนับเวลาเรียบร้อยแล้วจะเป็นการลบโทเค็นออกจากอินพุตเพลส แสดงในเหตุการณ์ removeToken โดยใช้กฎข้อที่ 7 แสดงตัวอย่างดังรูปที่ 3.20

```
event resetIndividualTransition
any t p
where
    @grd1 (t : dom(TIK)) & (STAT(t)=reset) & (postPlace={})
        & (POSTPL[{t}] <: s_PLACE)
    @grd2 (p : s_PLACE) & (p : PREPL[{t}])
then
    @act1 STAT(t) := hold
    @act2 TIK(t) := 0
end
event removeToken
any t pIn
where
    @grd1 (pIn : dom(s_MARKING)) & (pIn : PREPL[{t}])
        & (POSTPL[{t}] <: s_PLACE) & (s_MARKING(pIn)/=0)
    @grd2 !pp.(pp : prePlace=>prePlace <: s_PLACE
        & (s_MARKING(pp)>=WEIGHT))
    @grd3 STAT(t)=firing
then
    @act1 s_MARKING(pIn) := s_MARKING(pIn) - WEIGHT
end
event resetConcurrentTransition
any t
where
    @grd1 !pp.(pp : PREPL[{t}]=>PREPL[{t}] <: s_PLACE
        & (s_MARKING(pp)=0))
    @grd2 STAT(t)=firing
then
    @act1 STAT(t) := reset
end
```

กฎข้อที่ 7
เหตุการณ์ resetIndividualTransition
เหตุการณ์ removeToken
เหตุการณ์ resetConcurrentTransition

รูปที่ 3.20 ตัวอย่างการแปลงไทมด์เพทริเน็ตให้อยู่ในส่วนประกอบแมชชีนโดยใช้กฎการแปลงข้อที่ 7 สำหรับเหตุการณ์ resetIndividualTransition, เหตุการณ์ removeToken และ เหตุการณ์ resetConcurrentTransition

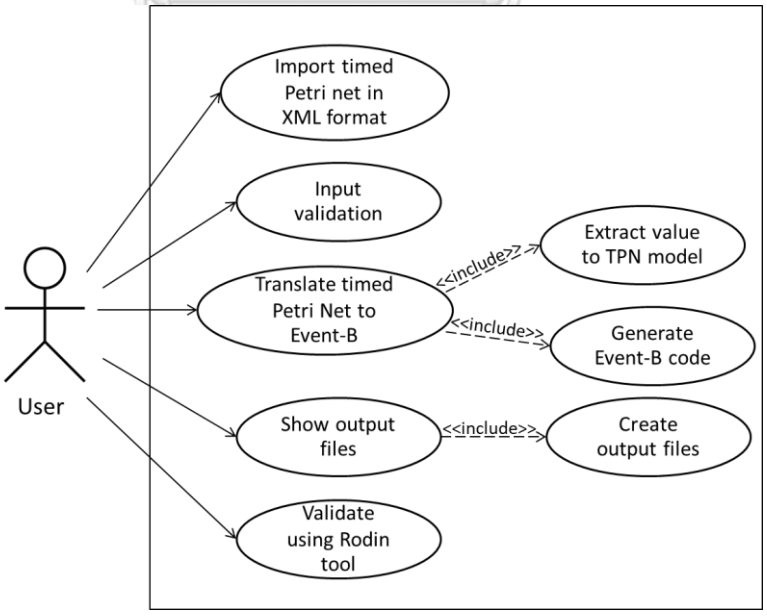
บทที่ 4
การออกแบบและการพัฒนาเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติ

4.1 การวิเคราะห์และออกแบบเครื่องมือ

การวิเคราะห์และออกแบบเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติ ผู้วิจัยจึงได้ยกเอาแผนภาพยูเคส (Use case diagram) เพื่อวิเคราะห์ขอบเขตการทำงานและฟังก์ชันในการทำงานของเครื่องมือ เมื่อวิเคราะห์เสร็จเรียบร้อย ผู้วิจัยได้ออกแบบส่วนต่อประสานผู้ใช้งานเพื่อแสดงให้เห็นขั้นตอนการพัฒนาเครื่องมือการแปลงและเพื่อแสดงส่วนประกอบต่อประสารของเครื่องมือ จากนั้นผู้วิจัยได้ออกแบบแผนภาพแพ็คเกจและแผนภาพคาส (Package diagram and Class diagram) เพื่อแสดงให้เห็นถึงความสัมพันธ์ของการทำงานของเครื่องมือการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี

4.1.1 แผนภาพยูสเคส

แผนภาพยูสเคสของเครื่องมือดังแสดงในรูปที่ 4.1 โดยผู้ใช้งานทำการเลือกแบบจำลองไทมด์เพทริเน็ตที่อยู่ในรูปแบบแฟ้มเอกสารเอกซ์เอ็มแอลที่ต้องการจะแปลง จากนั้นจะทำการแปลงแบบจำลองไทมด์เพทริเน็ตไปเป็นอีเวนท์บี ผู้ใช้งานต้องใส่ข้อมูลพารามิเตอร์ (Parameter) จะถูกแปลงไปเป็นส่วนประกอบของอีเวนท์บีที่อยู่ในรูปแบบของแฟ้มเอกสารข้อความ ซึ่งผู้ใช้สามารถคัดลอกผลลัพธ์การแปลงไปวางในเครื่องมือโรดินเพื่อทำการแดวงอีเวนท์ที่ได้จากการแปลง



รูปที่ 4.1 แผนภาพยูสเคสแสดงขอบเขตการทำงานของระบบ

จากรูปที่ 4.1 สามารถสรุปยูสเคสทั้งหมดของการพัฒนาเครื่องมือการแปลงได้ทั้งหมด 8 ยูสเคส สามารถแสดงรายละเอียดได้ในตารางที่ 4.1

ตารางที่ 4.1 แสดงยูสเคสไอดีของยูสเคสทั้งหมด

ลำดับ	ยูสเคสไอดี	ชื่อยูสเคส
1	UC-TPN2EB-01	Import timed Petri net in XML format
2	UC-TPN2EB-02	Input validation
3	UC-TPN2EB-03	Translate timed Petri Net to Event-B
4	UC-TPN2EB-04	Extract value to TPN model
5	UC-TPN2EB-05	Generate Event-B code
6	UC-TPN2EB-06	Create output files
7	UC-TPN2EB-07	Show output files
8	UC-TPN2EB-08	Validate using Rodin tool

รายละเอียดสำหรับแต่ละยูสเคสสามารถอธิบายได้ในตารางของคำอธิบายยูสเคส ซึ่งจะกล่าวในลำดับถัดไป

ตารางที่ 4.2 รายละเอียดยูสเคสการนำเข้าแบบจำลองไทมด์เพทริเน็ตในรูปแบบแฟ้มเอกสารเอกซ์เอ็มแอล

คำอธิบายยูสเคส	
Use case name : Import timed Petri net in XML format	**Use case ID :** UC-TPN2EB-01
Primary Business Actor : User	
Description : นำเข้าแบบจำลองไทมด์เพทริเน็ตที่อยู่ในรูปแบบแฟ้มเอกสารเอกซ์เอ็มแอล	
Pre–Condition : ระบุส่วนประกอบไทมด์เพทริเน็ตให้อยู่ในแฟ้มเอกสารเอกซ์เอ็มแอล	
Normal Flow : 1. ผู้ใช้เปิดเครื่องมือการแปลง 2. ผู้ใช้เลือกปุ่ม “Browse” และเลือกแฟ้มเอกสารนำเข้าเอกซ์เอ็มแอลที่ผู้ใช้ต้องการ 3. ผู้ใช้เลือกปุ่ม “OK” เพื่อตรวจสอบแฟ้มเอกนำเข้าสารเอกซ์เอ็มแอล	
Alternative Flow : 2a : ผู้ใช้ไม่เลือกปุ่ม “Browse” ในการเลือกแฟ้มเอกสารนำเข้าเอกซ์เอ็มแอล ผู้ใช้ระบุที่อยู่ของแฟ้มเอกสารเอกซ์เอ็มแอล 3a: ผู้ใช้ต้องการยกเลิกการทำงานของเครื่องมือการแปลง ผู้ใช้เลือกปุ่ม “Cancel” เพื่อยกเลิกการใช้งานเครื่องมือการแปลง	
Post–Condition : ระบบทำการตรวจสอบโครงสร้างและไวยากรณ์ของแฟ้มเอกสารนำเข้า	

ตารางที่ 4.3 รายละเอียดยูสเคสการตรวจสอบความถูกต้องของรูปแบบการนำเข้าแบบจำลองไทมด์เพทริเน็ต

คำอธิบายยูสเคส	
Use case name : Input validation	**Use case ID :** UC-TPN2EB-02
Primary Business Actor : TPN2EB_Translator	
Description : เพื่อทำการตรวจสอบความถูกต้องของไวยากรณ์และความถูกต้องของส่วนประกอบไทมด์เพทริเน็ตที่ผู้ใช้ระบุเข้ามา	
Pre–Condition : ผู้ใช้ระบุที่อยู่ของแฟ้มเอกสารเอกซ์เอ็มแอล	
Normal Flow : 1. อ่านแฟ้มเอกสารเอกซ์เอ็มแอลที่ผู้ใช้นำเข้ามา 2. ตรวจสอบไวยากรณ์เบื้องต้นของเอกซ์เอ็มแอล 3. ตรวจสอบแฟ้มเอกสารนำเข้าเป็นแบบจำลองไทมด์เพทริเน็ตที่อยู่ในรูปแบบของแฟ้มเอกสารเอกซ์เอ็มแอลหรือไม่ 4. ตรวจสอบไวยากรณ์ของการระบุส่วนประกอบไทมด์เพทริเน็ตนั้นสอดคล้องกับข้อกำหนดหรือไม่	
Alternative Flow : 2a1 : แฟ้มเอกสารนำเข้าผิดไวยากรณ์เบื้องต้นของเอกซ์เอ็มแอล ระบบแจ้งเตือนผู้ใช้เกี่ยวกับข้อผิดพลาดของแฟ้มเอกสารนำเข้าผิดไวยากรณ์เบื้องต้นของเอกซ์เอ็มแอล 3a1 : แฟ้มเอกสารที่นำเข้าไม่เป็นส่วนประกอบไทมด์เพทริเน็ต ระบบแจ้งเตือนผู้ใช้เกี่ยวกับข้อผิดพลาดของแฟ้มเอกสารนำเข้า 4a1 : พบข้อผิดพลาดของการระบุส่วนประกอบไทมด์เพทริเน็ตไม่สอดคล้องกับข้อกำหนด ระบบแจ้งเตือนผู้ใช้เกี่ยวกับข้อผิดพลาดของแฟ้มเอกสารนำเข้า	
Post–Condition เก็บข้อมูลและตรวจสอบแต่ละส่วนประกอบไทมด์เพทริเน็ตและแสดงส่วนประกอบไทมด์เพทริเน็ตในส่วนของส่วนต่อประสานเพื่อให้ผู้ใช้สามารถตรวจสอบได้	

ตารางที่ 4.4 รายละเอียดยูสเคสการแปลงแบบจำลองไทมด์เพทริเน็ตไปเป็นอีเวนท์บี

คำอธิบายยูสเคส	
Use case name : Translate timed Petri Net to Event-B	**Use case ID :** UC-TPN2EB-03
Primary Business Actor : User	
Description : การแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บี	
Pre–Condition: ผู้ใช้ระบุที่อยู่ของแฟ้มเอกสารเอกซ์เอ็มแอลและผ่านการตรวจความถูกต้องของโครงสร้างและไวยากรณ์ของแฟ้มเอกสารเอกซ์เอ็มแอล	
Normal Flow : 1. ผู้ใช้เลือกปุ่ม “Browse” เพื่อระบุที่อยู่ผลลัพธ์การแปลง 2. ผู้ใช้ระบุชื่อโปรเจกต์ที่ผู้ใช้ต้องการ 3. ผู้ใช้ระบุชื่อของส่วนประกอบบริบท 4. ผู้ใช้ระบุชื่อของส่วนประกอบแมชชีน	
Alternative Flow : 1a1: ผู้ใช้ไม่เลือกปุ่ม “Browse” เพื่อระบุที่อยู่ผลลัพธ์การแปลง ผู้ใช้ระบุที่อยู่ผลลัพธ์การแปลง 2a1: ผู้ใช้ระบุชื่อโปรเจกต์ซ้ำ ระบุแจ้งเตือนผู้ใช้เกี่ยวกับการพบโปรเจกต์ซ้ำในที่อยู่ของผลลัพธ์การแปลง	
Post–Condition : แปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นแฟ้มเอกสารข้อความ 2 เอกสารคือแฟ้มเอกสารข้อความของบริบทและแฟ้มเอกสารข้อความของแมชชีน ซึ่งเป็นส่วนประกอบหลักของอีเวนท์บี	

ตารางที่ 4.5 รายละเอียดยูสเคสการสกัดและเก็บข้องมูลส่วนประกอบไทมด์เพทริเน็ต

คำอธิบายยูสเคส	
Use case name : Extract value to TPN model	**Use case ID :** UC-TPN2EB-04
Primary Business Actor : TPN2EB_Translator	
Description : สกัดแฟ้มเอกสารเอกซ์เอ็มแอลเพื่อเก็บข้อมูลและตรวจสอบแต่ละส่วนประกอบไทมด์เพทริเน็ต	
Pre–Condition : ไม่พบข้อผิดพลาดของแฟ้มเอกสารนำเข้า	
Normal Flow : 1. อ่านป้าย <PLACES> เพื่อเก็บค่าของแอททริบิวต์ของแต่ละเพลส 2. อ่านป้าย <TRANSITIONS> เพื่อเก็บค่าของแอททริบิวต์ของแต่ละทรานซิชัน 3. อ่านป้าย <ARCS> เพื่อเก็บค่าของแอททริบิวต์ของแต่ละอินพุตเพลสหรือเอาต์พุตเพลส	
Alternative Flow : 1a1 : พบข้อผิดพลาดของ <PLACES> ระบบแจ้งเตือนผู้ใช้เกี่ยวกับข้อผิดพลาดของการระบุแบบจำลองไทมด์เพทริเน็ต 2a1 : พบข้อผิดพลาดของ < TRANSITIONS> ระบบแจ้งเตือนผู้ใช้เกี่ยวกับข้อผิดพลาดของการระบุแบบจำลองไทมด์เพทริเน็ต 3a1 : พบข้อผิดพลาดของ <ARCS> ระบบแจ้งเตือนผู้ใช้เกี่ยวกับข้อผิดพลาดของการระบุแบบจำลองไทมด์เพทริเน็ต	
Post–Condition : ดำเนินแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นแฟ้มเอกสารข้อความ สองเอกสารคือแฟ้มเอกสารข้อความของบริบทและแฟ้มเอกสารข้อความของแมชชีน ซึ่งเป็นส่วนประกอบหลักของอีเวนท์บี	

ตารางที่ 4.6 รายละเอียดยูสเคสการสร้างอีเวนท์บีโค้ด

คำอธิบายยูสเคส	
Use case name : Generate Event-B code	**Use case ID :** UC-TPN2EB-05
Primary Business Actor : TPN2EB_Translator	
Description : เพื่อทำการแปลงแต่ละส่วนประกอบของไทมด์เพทริเน็ตให้เป็นส่วนประกอบของอีเวนท์บี	
Pre–Condition : เลือกปุ่ม “Translate” และไม่พบข้อผิดพลาดในระหว่างการตรวจสอบความถูกต้องของแฟ้มเอกสารที่นำเข้า	
Normal Flow : 1. แปลงส่วนประกอบไทมด์เพทริเน็ตไปเป็นส่วนประกอบอีเวนท์บีโดยใช้กฎการแปลงที่นิยามขึ้นมา	
Alternative Flow : 1a1 : พบข้อผิดพลาดในระหว่างการแปลง ระบบแจ้งเตือนผู้ใช้ไม่สารมารถแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บี	
Post–Condition : เก็บส่วนประกอบของไทมด์เพทริเน็ตที่แปลงได้ไว้ในฐานข้อมูลภายในโปรแกรม	

ตารางที่ 4.7 รายละเอียดยูสเคสการเขียนผลลัพธ์การแปลงอีเวนท์บีให้อยู่ในแฟ้มเอกสารข้อความ

คำอธิบายยูสเคส	
Use case name : Create output file	**Use case ID :** UC-TPN2EB-06
Primary Business Actor : TPN2EB_Translator	
Description : เพื่อทำการสร้างแฟ้มเอกสารข้อความของอีเวนท์บี ซึ่งประกอบด้วยส่วนประกอบบริบทและแมชชีน	
Pre–Condition : เก็บส่วนประกอบของไทมด์เพทริเน็ตที่แปลงได้ไว้ในฐานข้อมูลภายในโปรแกรม	
Normal Flow : 1. สร้างแฟ้มเอกสารข้อความและดึงข้อมูลในฐานข้อมูลที่เก็บส่วนของบริบทเพื่อนำมาเขียนลงแฟ้มเอกสารข้อความที่เป็นส่วนของบริบท 2. บันทึกแฟ้มเอกสารข้อความของส่วนประกอบบริบท 3. สร้างแฟ้มเอกสารข้อความและดึงข้อมูลในฐานข้อมูลที่เก็บส่วนของแมชชีนเพื่อนำมาเขียนลงแฟ้มเอกสารข้อความที่เป็นส่วนของแมชชีน 4. บันทึกแฟ้มเอกสารข้อความของส่วนประกอบแมชชีน	
Alternative Flow : -	
Post–Condition : แจ้งเตือนการแปลงสำเร็จแล้ว	

ตารางที่ 4.8 รายละเอียดยูสเคสเพื่อแสดงหน้าต่างที่อยู่ของผลลัพธ์การแปลง

คำอธิบายยูสเคส	
Use case name : Show output file	**Use case ID :** UC-TPN2EB-07
Primary Business Actor : TPN2EB_Translator	
Other Interested Stakeholder : -	
Description : เพื่อแสดงผลลัพธ์การแปลงที่ถูกบันทึกไว้ในรูปแบบของแฟ้มเอกสารข้อความ	
Pre–Condition : เลือกปุ่ม Translate และไม่พบข้อผิดพลาดในระหว่างการตรวจสอบความถูกต้องของแฟ้มเอกสารที่นำเข้า	
Normal Flow : 1. เครื่องมือแสดงหน้าต่างสำหรับการแสดงผลลัพธ์การแปลง 2. ผู้ใช้เลือกปุ่ม “Open” เพื่อเปิดที่อยู่ของผลลัพธ์การแปลง 3. แสดงหน้าต่างที่อยู่ของผลลัพธ์การแปลง	
Alternative Flow : 2a1: ผู้ใช้ไม่ต้องการเปิดที่อยู่ของผลลัพธ์การแปลง ผู้ใช้เลือกปุ่ม “Cancel” เพื่อจบการทำงาน	
Post–Condition : แจ้งเตือนการแปลงสำเร็จแล้ว	

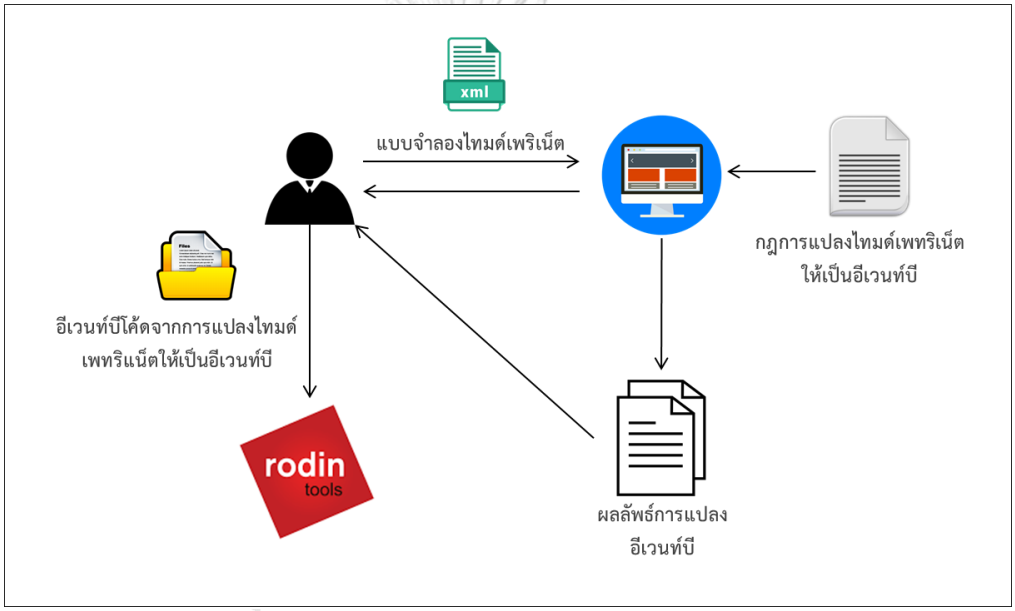
ตารางที่ 4.9 รายละเอียดยูสเคสการแสดงผลลัพธ์การแปลงอีเวนท์บี

คำอธิบายยูสเคส	
Use case name : Validate using Rodin tool	**Use case ID :** UC-TPN2EB-08
Primary Business Actor : User	
Other Interested Stakeholder : -	
Description : เพื่อแสดงอีเวนท์บีที่ได้จากการแปลงบนเครื่องมือโรดิน	
Pre–Condition : การแปลงสำเร็จแล้ว	
Normal Flow : 1. ผู้ใช้งานเปิดเครื่องมือโรดิน 2. สร้างส่วนประกอบบริบท 3. คัดลอกผลลัพธ์การแปลงจากแฟ้มเอกสารข้อความในส่วนของบริบทและนำไปวางในส่วนประกอบของบริบทที่สร้างขึ้น 4. สร้างส่วนประกอบแมชชีน 5. คัดลอกผลลัพธ์การแปลงจากแฟ้มเอกสารข้อความในส่วนของแมชชีนและนำไปวางในส่วนประกอบแมชชีนที่สร้างขึ้น	
Alternative Flow : 3a1 : ชื่อส่วนประกอบของบริบทไม่ตรงกับชื่อบริบทที่ได้จากผลลัพธ์การแปลง เปลี่ยนให้ชื่อสอดคล้องกับบริบทที่สร้างภายในเครื่องมือโรดิน 5a1 : ชื่อส่วนประกอบของบริบทไม่ตรงกับชื่อบริบทที่ได้จากผลลัพธ์การแปลง เปลี่ยนให้ชื่อสอดคล้องกับแมชชีนทที่สร้างภายในเครื่องมือโรดิน	
Post–Condition : แจ้งเตือนการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีสำเร็จแล้ว	

4.2 การออกแบบภาพรวมการทำงานของเครื่องมือการแปลงไทมด์เพริเน็ตให้เป็นอีเวนท์

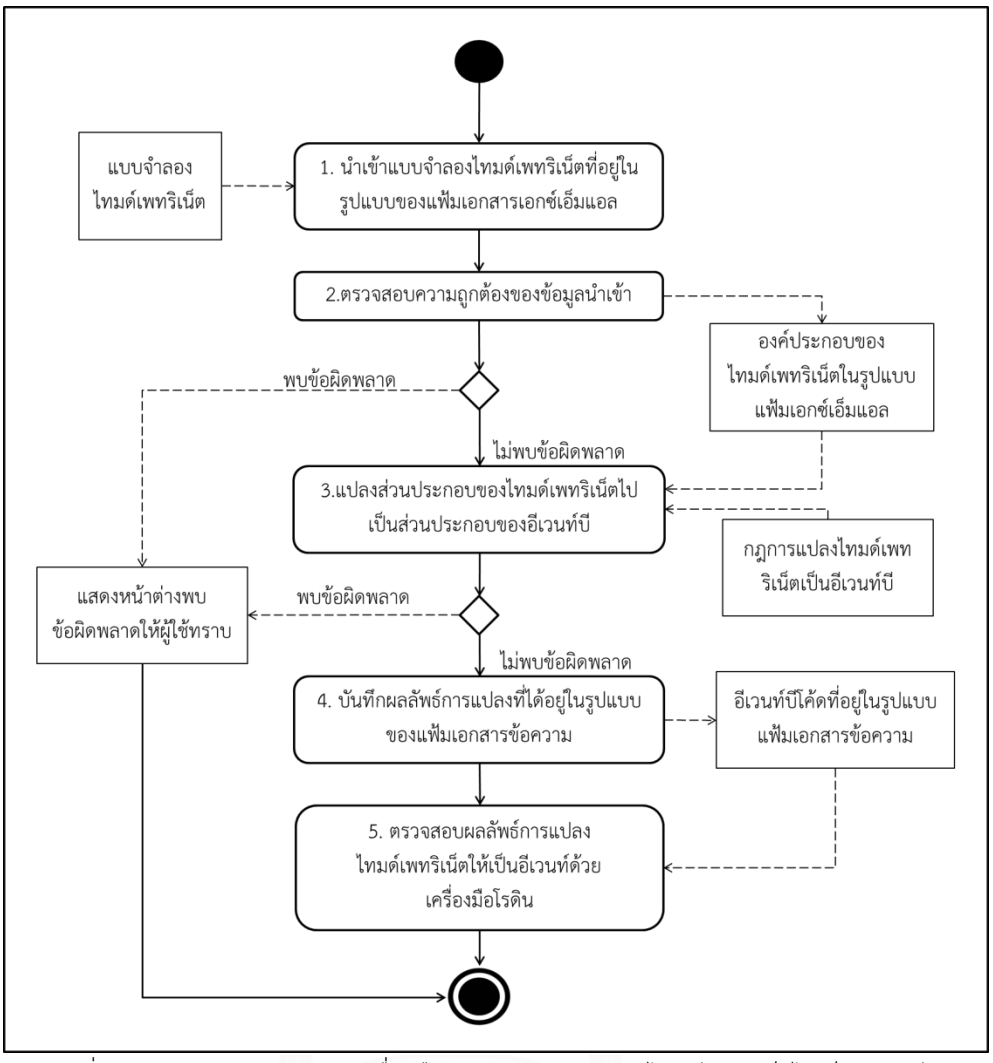
เครื่องมืออัตโนมัติในการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีจะพัฒนาภายใต้สภาพแวดล้อมโดยใช้โปรแกรมภาษาจาวา (Java Programming Language) ภายใต้โปรแกรมอีคลิปส์ (Eclipse) และอยู่ในรูปแบบของวินโดว์แอพพลิเคชันโดยเครื่องมือนี้จะทำการแปลงส่วนประกอบของไทมด์เพทริเน็ตไปเป็นส่วนประกอบของอีเวนท์บี โดยใช้กฎในการแปลงที่ผู้วิจัยกำหนดขึ้น

เริ่มต้นการทำงานจากผู้ใช้งานนำเข้าแบบจำลองไทมด์เพทริเน็ตไปในเครื่องมือการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี จากนั้นเครื่องมือทำการแปลงโดยใช้กฎการแปลงที่ได้นิยามไว้เพื่อให้ได้ผลลัพธ์การแปลงอีเวนท์บี จากนั้นผู้ใช้นำผลลัพธ์การแปลงอีเวนท์บีไปตรวจสอบผ่านเครื่องมือโรดิน แสดงแผนภาพกิจกรรมในรูปที่ 4.2



รูปที่ 4.2 แนวคิดการทำงานของเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บี

จากรูปที่ 4.2 แสดงแผนภาพกิจกรรมรวมการดำเนินงานของเครื่องมือการแปลงแบบจำลองไทมด์เพทริเน็ตไปเป็นอีเวนท์บี ดังรูปที่ 4.3



รูปที่ 4.3 แผนภาพกิจกรรมของเครื่องมือการแปลงแบบจำลองไทมด์เพทริเน็ตไปเป็นอีเวนท์บี

สามารถแสดงหลักการทำงานของเครื่องมือ โดยมีรายละเอียดดังต่อไปนี้

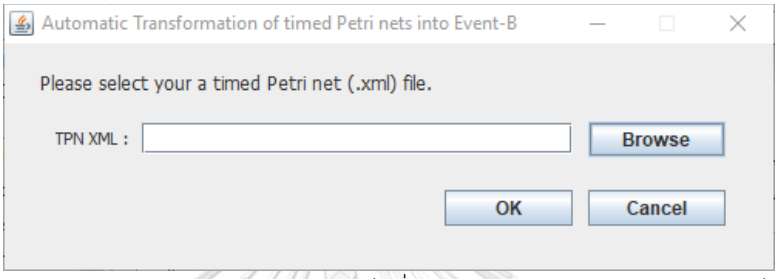
1) ผู้ใช้งานนำเข้าแบบจำลองไทมด์เพทริเน็ตที่อยู่ในรูปแบบของแฟ้มเอกสารเอกซ์เอ็มแอล
2) เครื่องมือการแปลงตรวจสอบความถูกต้องของข้อมูลนำเข้า
3) เครื่องมือใช้กฎในการแปลงส่วนประกอบของไทมด์เพทริเน็ตไปเป็นส่วนประกอบของอีเวนท์บี
4) เครื่องมือการแปลงบันทึกผลลัพธ์การแปลงที่ได้อยู่ในรูปแบบของแฟ้มเอกสารข้อความ (Text File) และสามารถคัดลอกไปวางในเครื่องมือโรดิน
5) ผู้ใช้งานตรวจสอบผลลัพธ์การแปลงอีเวนท์บีด้วยเครื่องมือโรดิน

4.2.1 การออกแบบส่วนต่อประสานผู้ใช้งาน

ส่วนต่อประสานผู้ใช้ออกแบบมาเพื่อรองรับการนำข้อมูลนำเข้า โดยมีรายละเอียดหน้าจอหลัก หน้าจอการหาที่อยู่ของแฟ้มเอกสารเอกซ์เอ็มแอล

1) หน้าจอนำเข้าจำลองไทมด์เพทริเน็ตที่อยู่ในรูปแบบของแฟ้มเอกสารเอกซ์เอ็มแอล

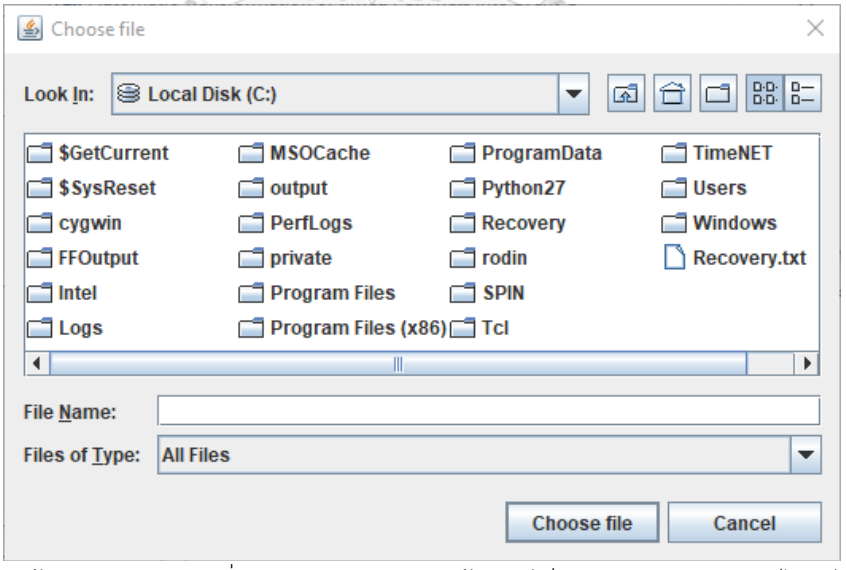
หน้าจอนำเข้าของแบบจำลองไทมด์เพทริเน็ตเป็นส่วนในการเริ่มต้นการทำงานของเครื่องมือและเป็นหน้าจอที่ใช้ในการเรียกหน้าจอต่างๆ ขึ้นมาใช้งานฟังก์ชันของเครื่องมือ แสดงในรูปที่ 4.4



รูปที่ 4.4 หน้าจอนำเข้าจำลองไทมด์เพทริเน็ตที่อยู่ในรูปแบบของแฟ้มเอกสารเอกซ์เอ็มแอล

2) หน้าจอสำหรับหาที่อยู่ของแฟ้มเอกสารนำเข้าเอกซ์เอ็มแอล

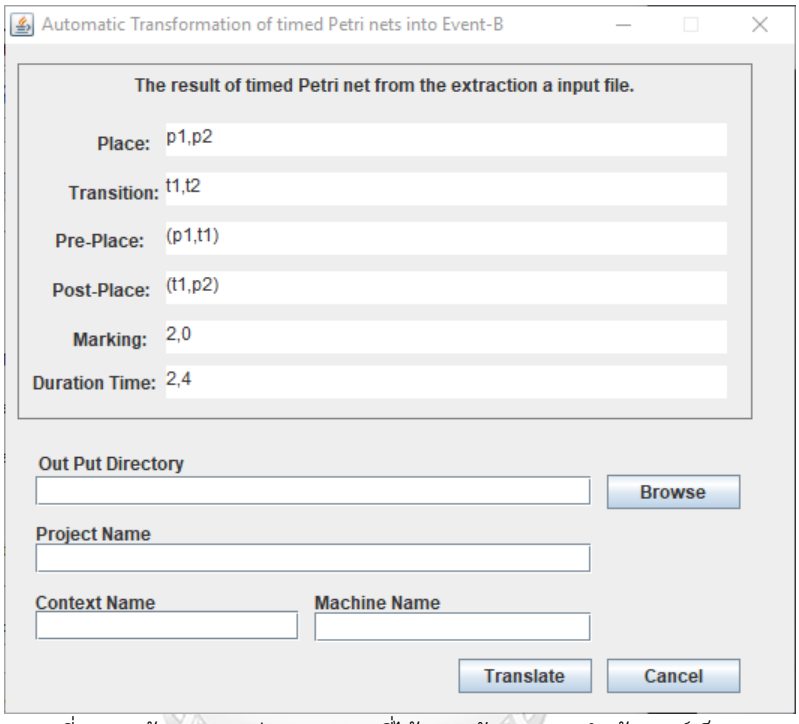
หน้าจอสำหรับการหาที่อยู่ของแฟ้มเอกสารนำเข้าเอกซ์เอ็มแอลของส่วนประกอบไทมด์เพทริเน็ต ซึ่งเป็นข้อมูลนำเข้าของเครื่องมือการแปลง แสดงในรูปที่ 4.5



รูปที่ 4.5 หน้าจอสำหรับการหาที่อยู่ของแฟ้มเอกสารนำเข้าเอกซ์เอ็มแอลของส่วนประกอบไทมด์เพทริเน็ต

3) หน้าจอแสดงส่วนประกอบที่ได้จากแฟ้มเอกสารนำเข้าเอกซ์เอ็มแอล

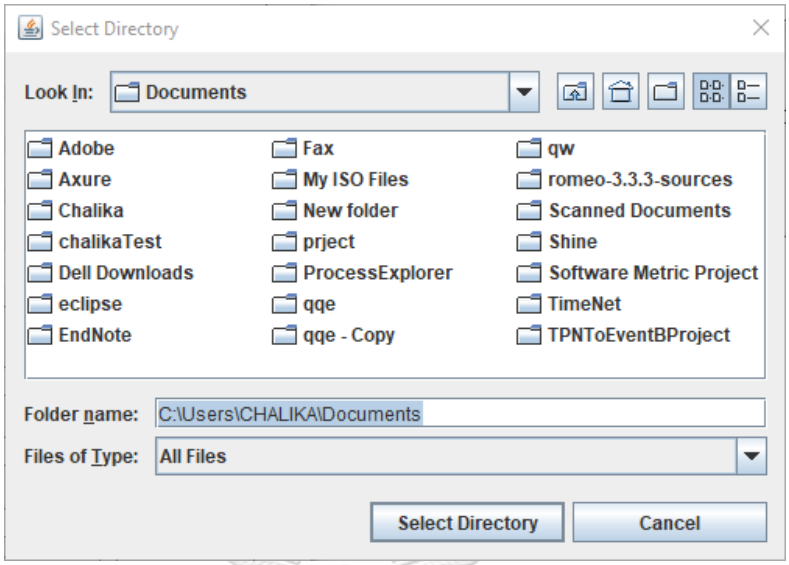
เมื่อผู้ใช้เลือกแฟ้มเอกสารเอกซ์เอ็มแอล ระบบจะแสดงหน้าจอสำหรับแสดงส่วนประกอบไทมด์เพทริเน็ตที่ผู้ใช้ระบุเข้ามาและถ้าหากผู้ใช้ต้องการดำเนินการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี ผู้ใช้สามารถระบุที่อยู่ของผลลัพธ์การแปลง, ชื่อโปรเจกต์, ชื่อของส่วนประกอบบริบทและชื่อของส่วนประกอบแมชชีน แสดงในรูปที่ 4.6



รูปที่ 4.6 หน้าจอแสดงส่วนประกอบที่ได้จากแฟ้มเอกสารนำเข้าเอกซ์เอ็มแอล

4) หน้าจอสำหรับเลือกที่อยู่ของผลลัพธ์การแปลง

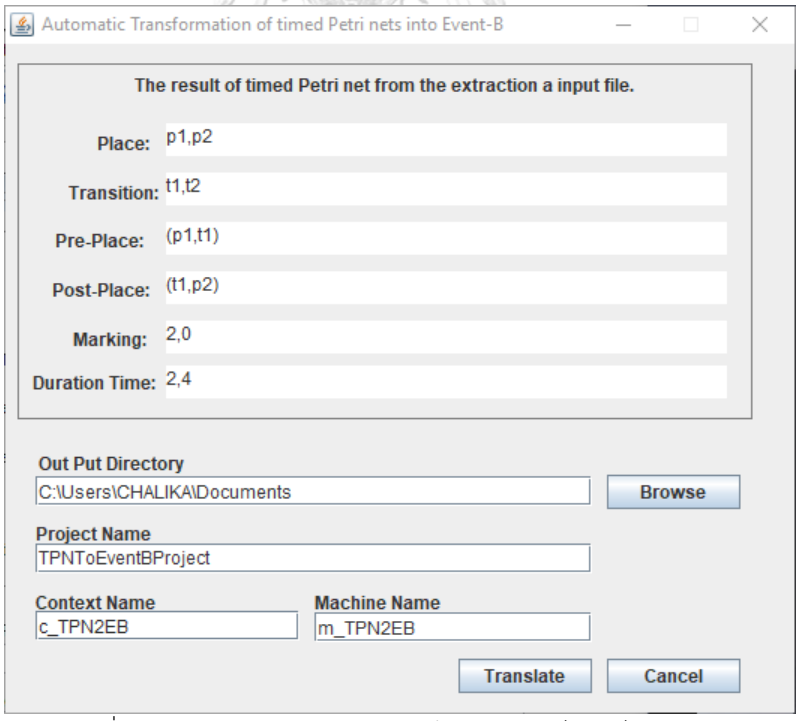
หน้าจอการระบุตำแหน่งของผลลัพธ์การแปลง ใช้สำหรับระบุตำแหน่งที่ผู้ใช้ต้องการให้ผลลัพธ์การแปลงไปอยู่ในตำแหน่งที่ต้องการ เมื่อทำการแปลงเสร็จเรียบร้อยแล้วเครื่องมือการแปลงจะทำการบันทึกไปยังตำแหน่งที่ผู้ใช้ระบุ เพื่อให้ง่ายต่อการค้นหา แสดงในรูปที่ 4.7



รูปที่ 4.7 หน้าจอการระบุตำแหน่งสำรับการบันทึกแฟ้มเอกสารข้อความของผลลัพธ์การแปลง

5) หน้าสำหรับดำเนินการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี

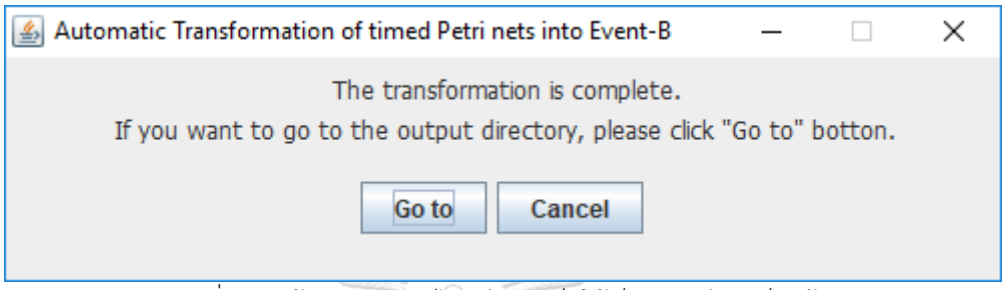
เมื่อผู้ใช้ต้องการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี ซึ่งผู้ใช้ต้องระบุที่อยู่ของผลลัพธ์การแปลง, ชื่อโปรเจกต์, ชื่อของส่วนประกอบบริบทและชื่อของส่วนประกอบแมชชีน แสดงในรูปที่ 4.8



รูปที่ 4.8 หน้าสำหรับดำเนินการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี

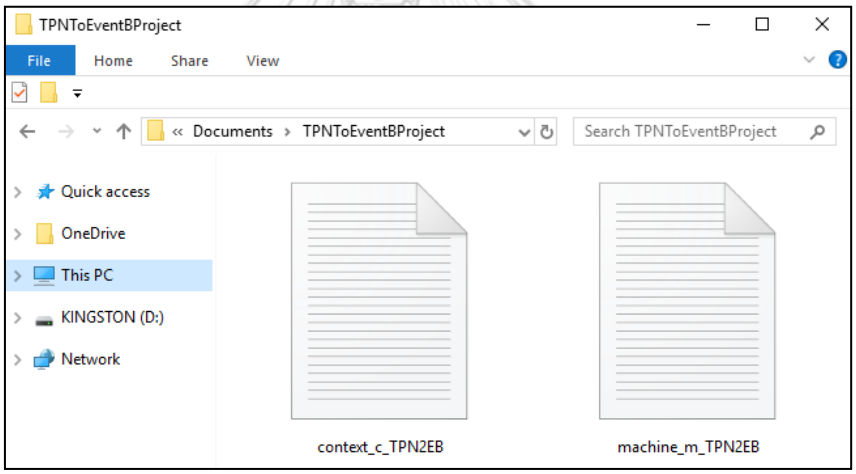
6) หน้าจอการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีสำเร็จแล้ว

เมื่อดำเนินการแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บีสำเร็จ เครื่องมือจะแสดงหน้าการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีสำเร็จแล้ว แสดงในรูปที่ 4.9 หากผู้ใช้ต้องการเปิดที่อยู่ของผลลัพธ์การแปลง ซึ่งผู้ใช้สามารถเลือกกดปุ่ม “Go to” หากต้องการจบการทำงานผู้ใช้สามารถเลือกกดปุ่ม “Cancel”



รูปที่ 4.9 หน้าจอการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีสำเร็จแล้ว

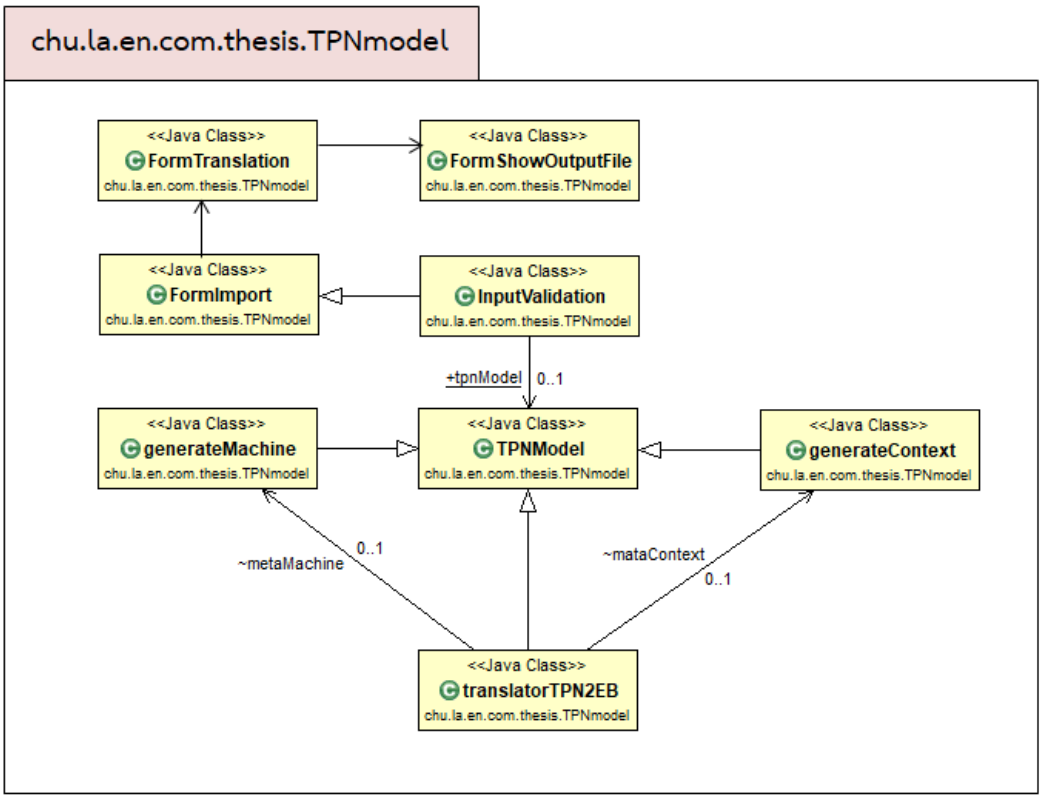
เมื่อผู้ใช้เลือกปุ่ม “Go to” เครื่องมือจะทำการเปิดที่อยู่ของผลลัพธ์การแปลง แสดงในรูปที่ 4.10



รูปที่ 4.10 หน้าต่างที่อยู่ของผลลัพธ์การแปลง

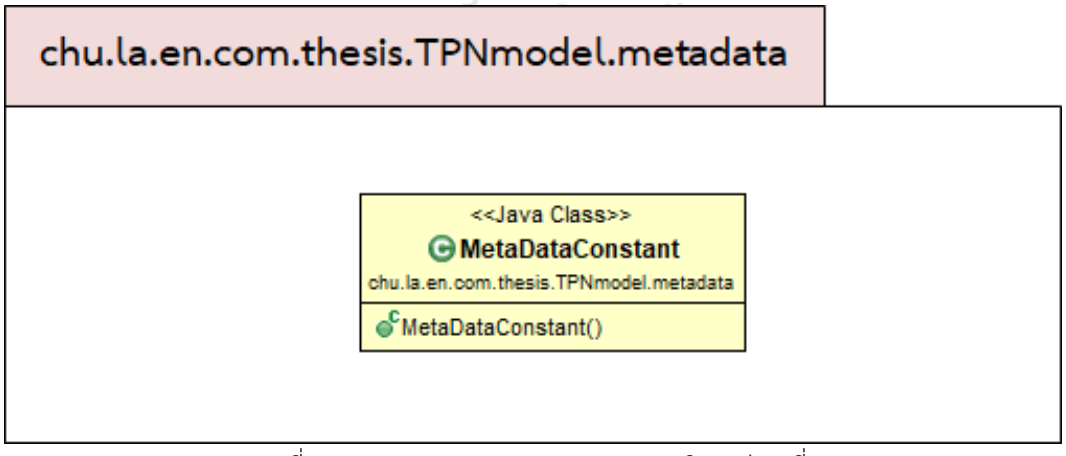
4.2.2 แผนภาพแพ็คเกจและแผนภาพคลาส (Package diagram and Class diagram)

แผนภาพคลาสจำลองเพื่อให้เห็นถึงการออกแบบและแสดงความสัมพันธ์ระหว่างคลาสต่างๆ ของวิธีการพัฒนาระบบ แสดงความสัมพันธ์ของแต่ละคลาสได้ซึ่งแบ่งออกเป็น 2 แพ็คเก็จ จากรูปที่ 4.11 แสดงแพ็คเกจ “chu.la.en.com.thesis.TPNmodel” ซึ่งเป็นแพ็คเกจสำหรับพัฒนาเครื่องมือการแปลงและแสดงแผนภาพคลาสของการพัฒนาเครื่องมือการแปลง



รูปที่ 4.11 แผนภาพคลาสของการพัฒนาเครื่องมือการแปลง

จากรูปที่ 4.12 แสดงแพ็คเกจ “chu.la.en.com.thesis.TPNmodel.metadata” ซึ่งเป็นแพ็คเกจสำหรับนิยามค่าคงที่ ที่ใช้สำหรับการแปลงและแสดงแผนภาพคลาสของการนิยามค่าคงที่

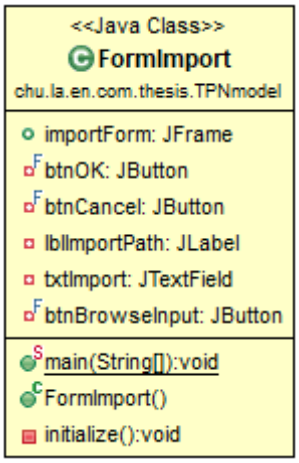


รูปที่ 4.12 แผนภาพคลาสของคลาสของการนิยามค่าคงที่

อธิบายความหมายของคลาสหลักได้ดังต่อไปนี้

1) คลาส FormImport

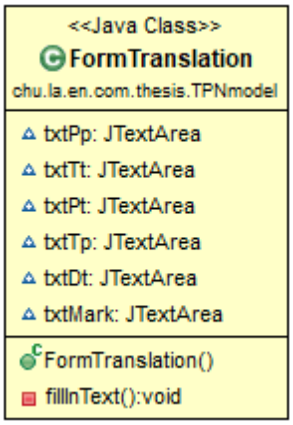
คลาส FormImport เป็นคลาสที่เกี่ยวข้องกับส่วนต่อประสานกับผู้ใช้ เมื่อผู้ใช้เริ่มต้นใช้งานเครื่องมือการแปลง แสดงรายละเอียดของคลาสดังรูปที่ 4.13



รูปที่ 4.13 รายละเอียดคลาส FormImport

2) คลาส FormTranslation

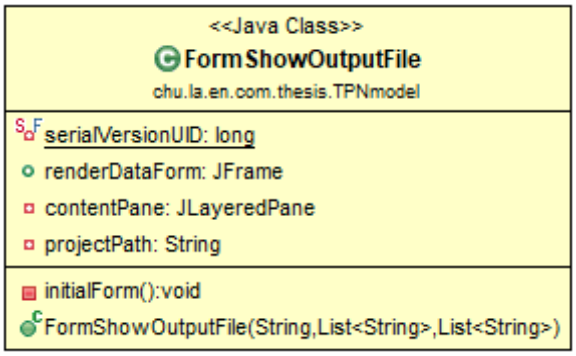
คลาส FormTranslation เป็นคลาสที่เกี่ยวข้องกับส่วนต่อประสานทำหน้าที่แปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บี แสดงรายละเอียดของคลาสดังรูปที่ 4.14



รูปที่ 4.14 รายละเอียดคลาส FormTranslation

3) คลาส FormShowOutputFIle

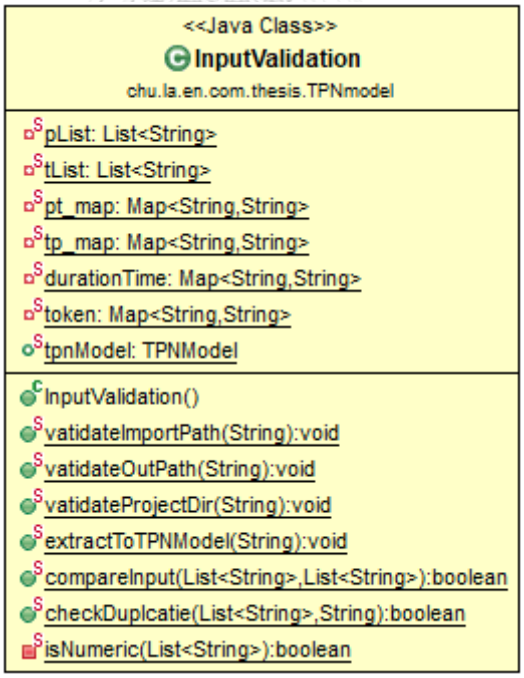
คลาส FormShowOutputFIle เป็นคลาสที่เกี่ยวข้องกับส่วนต่อประสานทำหน้าที่แสดงที่อยู่ของผลลัพธ์การแปลง แสดงรายละเอียดของคลาสดังรูปที่ 4.15



รูปที่ 4.15 รายละเอียดคลาส FormShowOutputFIle

4) คลาส InputValidation

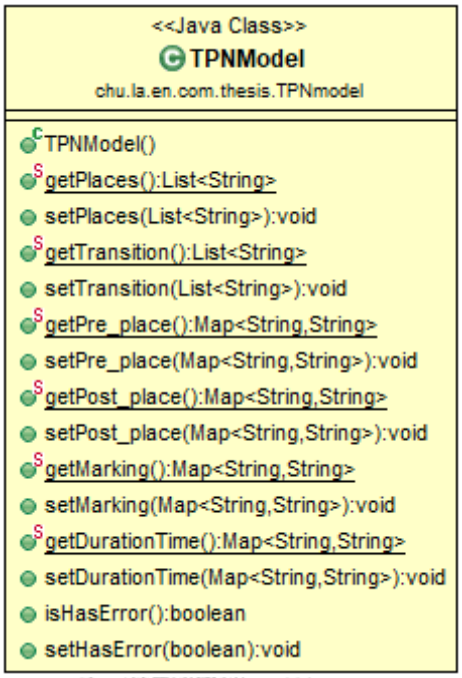
คลาส InputValidation เป็นคลาสตรวจสอบความถูกต้องของข้อมูลนำเข้า แสดงรายละเอียดของคลาสในรูปที่ 4.16



รูปที่ 4.16 รายละเอียดคลาส InputValidation

5) คลาส TPNModel

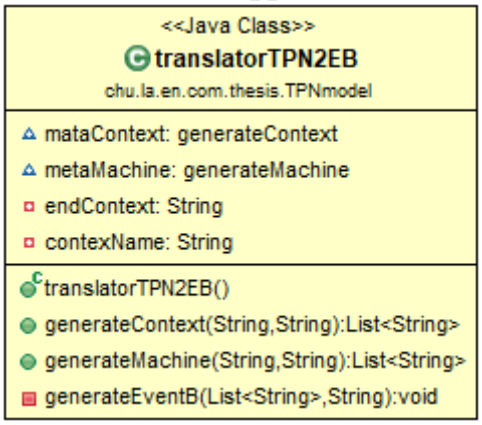
คลาส TPNModel เป็นคลาสที่ของวัตถุที่เก็บรวบรวมส่วนประกอบทั้งหมดของไทมด์เพทริเน็ตและสามารถให้คลาสอื่นเรียกใช้ขอมูลของวัตถุได้ แสดงรายละเอียดของคลาสในรูปที่ 4.17



รูปที่ 4.17 รายละเอียดคลาส TPNModel

6) คลาส translatorTPN2EB

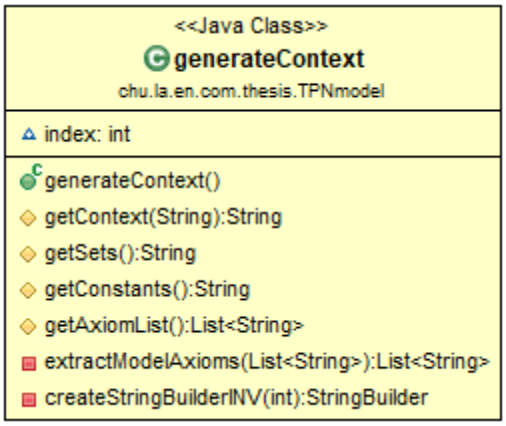
คลาส translatorTPN2EB เป็นคลาสสำหรับแปลงไทมด์เพริเน็ตไปเป็นอีเวนท์บี แสดงรายละเอียดของคลาสรูปที่ 4.18



รูปที่ 4.18 รายละเอียดคลาส translatorTPN2EB

7) คลาส generateContext

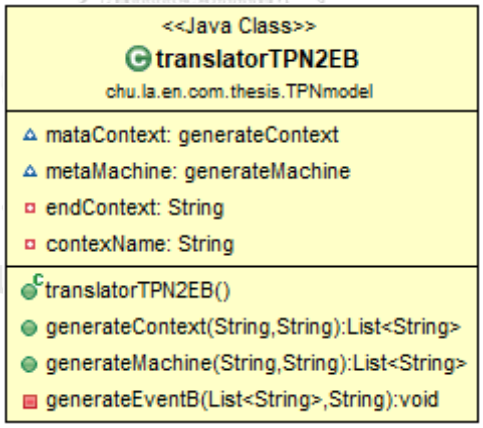
คลาส generateContext เป็นคลาสสำหรับแปลงไทมด์เพริเน็ตไปเป็นส่วนประกอบบริบทสำหรับอีเวนท์บี แสดงรายละเอียดของคลาสในรูปที่ 4.19



รูปที่ 4.19 รายละเอียดคลาส generateContext

8) คลาส tranlatorTPN2EB

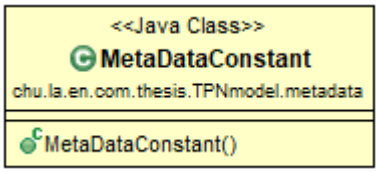
คลาส tranlatorTPN2EB เป็นคลาสสำหรับแปลงไทมด์เพริเน็ตไปเป็นส่วนประกอบแมชชีนสำหรับอีเวนท์บี แสดงรายละเอียดของคลาสในรูปที่ 4.20



รูปที่ 4.20 รายละเอียดคลาส tranlatorTPN2EB

9) คลาส MetaDataConstant

คลาส MetaDataConstant เป็นคลาสที่ทำหน้าที่นิยามตัวแปลงค่าคงที่สำหรับใช้ภายในเครื่องมือการแปลง แสดงรายละเอียดของคลาสในรูปที่ 4.21



รูปที่ 4.21 รายละเอียดคลาส MetaDataConstant

4.3 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ

สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมืออัตโนมัติสำหรับการแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บี มีสภาพแวดล้อมทางด้านฮาร์ดแวร์ (Hardware) และทางด้านซอฟต์แวร์ (Software)

4.3.1 ฮาร์ดแวร์

1) เครื่องคอมพิวเตอร์โน้ตบุ๊ก หน่วยประมวลผลอินเทลคอร์ไอเซเวน 2.70 กิกะเฮิร์ท (Intel Core i7 2.70 GHZ)
2) หน่วยความจำสำรอง (RAM) 8.0 กิกะไบต์ (8.0 GB)
3) ฮาร์ดดิสก์ (Harddisk) 250 กิกะไบต์ (250 GB)

4.3.2 ซอฟต์แวร์

1) ระบบปฏิบัติการ (Operating System) ไมโครซอฟต์วินโดว์เซเวน โพรเฟซชันแนว เซอร์วิสแพค 1 (Microsoft Windows 7 Professional Service Pack 1)
2) เครื่องมือที่ใช้พัฒนา
    - โปรแกรมอีคลิปส์ เวอร์ชัน ลูน่า 4.4.2 (Eclipse version Luna 4.4.2)
3) ภาษาที่ใช้พัฒนา
    - ภาษาจาวา (Java language)

บทที่ 5
การทดสอบเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติ

การทดสอบมีวัตถุประสงค์ในการทดสอบเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติและกฎการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บี ซึ่งผู้วิจัยได้สร้างขึ้นมานั้นสามารถทำงานได้ถูกต้องและตรงตามข้อกำหนดที่ได้ออกแบบไว้

5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ

สำหรับกาทดสอบเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติ ผู้วิจัยเลือกใช้คอมพิวเตอร์และโปรแกรมประยุกต์ที่ใช้ทดสอบเครื่องมืออัตโนมัติสำหรับการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บี โดยมีรายละเอียดดังต่อไปนี้

1) เครื่องคอมพิวเตอร์โน้ตบุ๊ก หน่วยประมวลผลอินเทลคอร์ไอเซเวน 2.70 กิกะเฮิร์ท (Intel Core i7 2.70 GHZ)
2) หน่วยความจำสำรอง (RAM) 8.0 กิกะไบต์ (8.0 GB)
3) ฮาร์ดดิสก์ (Harddisk) 250 กิกะไบต์ (250 GB)
4) เครื่องมือโรดิน เวอร์ชัน 3.3.0

5.2 แนวทางการทดสอบ

การทดสอบเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติ มีแนวทางการทดสอบดังต่อไปนี้

1) เริ่มต้นจากการระบุส่วนประกอบไทมด์เพทริเน็ตให้อยู่ในแฟ้มเอกสารเอกซ์เอ็มแอล
2) เครื่องมืออัตโนมัติแปลส่วนประกอบไทมด์เพทริเน็ตไปเป็นส่วนประกอบบริบทและส่วนประกอบแมชชีน ซึ่งเป็นส่วนประกอบของอีเวนท์บีได้อย่างถูกต้อง
3) เปิดเครื่องมือโรดินและสร้างส่วนประกอบบริบทและส่วนประกอบแมชชีน จากนั้นคัดลอกผลลัพธ์การแปลงไปวางในส่วนประกอบที่สร้างขึ้น
4) ทำการทวนสอบผลลัพธ์การแปลงอีเวนท์บี

5.3 การทดสอบและประเมินผลระบบ

สำหรับการทดสอบกฎการแปลงและเครื่องมือการแปลงแบบจำลองไทมด์เพทริเน็ตไปเป็นอีเวนท์บี ซึ่งจะแบ่งออกเป็น 2 กลุ่ม มีดังต่อไปนี้

1) ประเมินผลการทำงานของเครื่องมือการแปลงโดยใช้กรณีทดสอบ สำหรับการตรวจสอบไวยากรณ์และความหมายของข้อมูลนำเข้าแฟ้มเอกสารเอกซ์เอ็มแอลของแบบจำลองไทมด์เพทริเน็ต มีรายละเอียดย่อย คือ

1.1) ตรวจสอบไวยากรณ์และความหมายของแบบจำลองไทมด์เพทริเน็ต

1.2) ตรวจสอบความครบถ้วนขององค์ประกอบของไทมด์เพทริเน็ตที่อยู่ในผลลัพธ์การแปลงอีเวนท์บี

2) ประเมินผลเครื่องมือโดยใช้กรณีศึกษาสำหรับตรวจสอบคุณสมบัติของผลลัพธ์การแปลงอีเวนท์บี โดยใช้เครื่องมือโรดิน ซึ่งจะใช้ 3 กรณีศึกษา คือ ระบบไฟจราจร (Traffic light system) ระบบการทำงานแบบพร้อมกัน (Concurrent System) และระบบห่วงโซ่อุปทาน (Supply Chain) โดยจะมีการคุณสมบัติดังต่อไปนี้

2.1) ตรวจสอบคุณสมบัติด้านความปลอดภัยของระบบ (Safety Property)

2.2) ตรวจสอบหาภาวะติดตายของระบบ (Deadlock)

2.3) ตรวจสอบเวลาการยิงสำหรับแต่ละทรานซิชันในอีเวนท์บีต้องสอดคล้องกับนิยามของไทมด์เพทริเน็ต (Firing sequence of each transition)

5.3.1 ประเมินผลการทำงานของเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติ โดยใช้กรณีทดสอบ สำหรับตรวจสอบไวยากรณ์และความหมายของข้อมูลนำเข้า

สำหรับการตรวจสอบไวยากรณ์และความหมายของข้อมูลนำเข้าแฟ้มเอกสารเอกซ์เอ็มแอลของแบบจำลองไทมด์เพทริเน็ต ผู้วิจัยได้เลือกกรณีทดสอบย่อยสำหรับการตรวจสอบไวยากรณ์และความหมายของแบบจำลองไทมด์เพทริเน็ตทั้งหมด 7 กรณีทดสอบย่อย

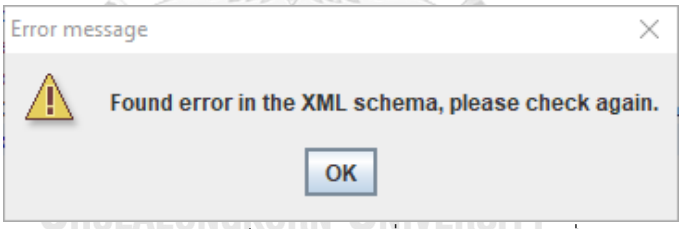
1) ตรวจสอบไวยากรณ์และความหมายของแบบจำลองไทมด์เพทริเน็ต มีกรณีทดสอบย่อยดังต่อไปนี้

1.1) ตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีที่ไม่ใช่แบบจำลองไทมด์เพทริเน็ต กรณีนี้จะเกิดขึ้นเมื่อผู้ใช้นำเข้าข้อมูลเอกสารเอกซ์เอ็มแอลที่ไม่ใช่โครงสร้างของไทมด์เพทริเน็ต แสดงในรูปที่ 5.1

```xml
<?xml version="1.0" encoding="UTF-8"?>
<net gridActive="true" id="0" netclass="SCPN" sharpEdges="false"
  xmlns="http://pdv.cs.tu-berlin.de/TimeNET/schema/SCPN"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" etc/schemas/SCPN.xsd">
  <place capacity="0" id="0.0" initialMarking="" queue="Random"
    tokentype="int" type="node" watch="false">
    <graphics orientation="0" x="470" y="90"/>
    <label id="0.0.0" text="P0" type="text">
      <graphics x="-10" y="-40"/>
    </label>
  </place>
  <place capacity="0" id="0.3" queue="Random" tokentype="int"
    type="node" watch="false">
    <graphics orientation="0" x="360" y="300"/>
    <label id="0.3.0" text="P1" type="text">
      <graphics x="-10" y="-40"/>
    </label>
  </place>
  <place capacity="0" id="0.4" queue="Random" tokentype="int"
    type="node" watch="false">
    <graphics orientation="0" x="530" y="300"/>
    <label id="0.4.0" text="P2" type="text">
      <graphics x="-10" y="-40"/>
    </label>
  </place>
  <timedTransition id="0.1" serverType="ExclusiveServer"
    specType="Automatic" takeFirst="false" timeFunction="EXP(1.0)"
    type="node" watch="false">
    <graphics orientation="0" x="370" y="180"/>
    <label id="0.1.0" text="T0" type="text">
      <graphics x="-10" y="-40"/>
    </label>
```

รูปที่ 5.1 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีที่นำเข้าผิดเอกสารที่ไม่ใช่แบบจำลองไทมด์เพทริเน็ต

ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีที่นำเข้าผิดเอกสารที่ไม่ใช่แบบจำลองไทมด์เพทริเน็ต แสดงดังรูปที่ 5.2



รูปที่ 5.2 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีที่นำเข้าผิดเอกสารที่ไม่ใช่แบบจำลองไทมด์เพทริเน็ต

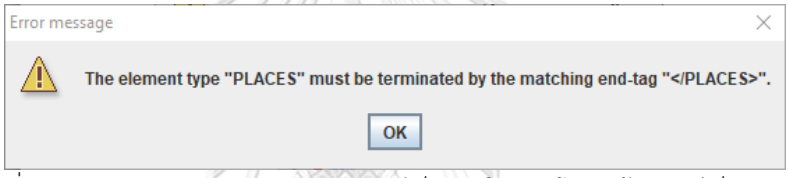
1.2) กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีโครงสร้างเอกซ์เอ็มแอลผิด โดยที่เอกสารนำเข้าเอกซ์เอ็มแอลไม่มีการระบุของป้ายระบุปิด </PLACES> แสดงในรูปที่ 5.3

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<TPN>
    <PLACES>
        <PLACE id="PL_p1" name="p1" marking="2" type="node"/>
        <PLACE id="PL_p2" name="p2" marking="0" type="node"/>

    <TRANSITIONS>
        <TRANSITION id="TR_t1" name="t1" durationTime="2"  type="node"/>
        <TRANSITION id="TR_t2" name="t2" durationTime="4"  type="node"/>
    </TRANSITIONS>
    <ARCS>
        <ARC id="inputPlace" fromNode="p1" toNode="t1" type="node"/>
        <ARC id="outputPlace" fromNode="t1" toNode="p2" type="node"/>
    </ARCS>
</TPN>
```

รูปที่ 5.3 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีโครงสร้างเอกซ์เอ็มแอลผิด

ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีโครงสร้างเอกซ์เอ็มแอลผิด แสดงดังรูปที่ 5.4



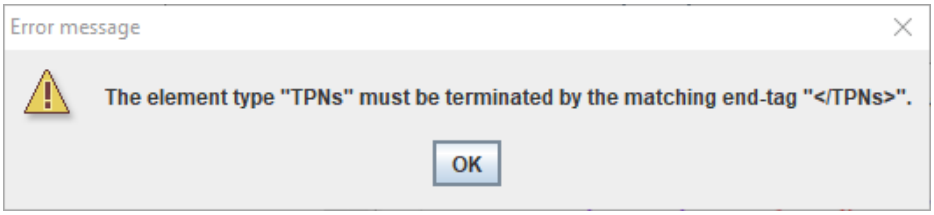
รูปที่ 5.4 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลในกรณีโครงสร้างเอกซ์เอ็มแอลผิด

1.3) ตรวจสอบไวยากรณ์ของไทมด์เพทริเน็ตในกรณีที่ผู้ใช้เปลี่ยนแปลงโครงสร้างของเอกซ์เอ็มแอล โดยที่เอกสารนำเข้าเอกซ์เอ็มแอลมีการระบุของป้ายระบุปิดที่ไม่ตรงกับป้ายระบุเปิด <TPNS> ... </TPN> แสดงในรูปที่ 5.5

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<TPNs>
    <PLACES>
        <PLACE id="PL_p1" name="p1" marking="2" type="node"/>
        <PLACE id="PL_p2" name="p2" marking="0" type="node"/>
    </PLACES>
    <TRANSITIONS>
        <TRANSITION id="TR_t1" name="t1" durationTime="2"  type="node"/>
        <TRANSITION id="TR_t2" name="t2" durationTime="4"  type="node"/>
    </TRANSITIONS>
    <ARCS>
        <ARC id="inputPlace" fromNode="p1" toNode="t1" type="node"/>
        <ARC id="outputPlace" fromNode="t1" toNode="p2" type="node"/>
    </ARCS>
</TPN>
```

รูปที่ 5.5 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้เปลี่ยนแปลงโครงสร้างของเอกซ์เอ็มแอล

ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้เปลี่ยนแปลงโครงสร้างของเอกซ์เอ็มแอล แสดงดังรูปที่ 5.6



รูปที่ 5.6 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้เปลี่ยนแปลงโครงสร้างของเอกซ์เอ็มแอล

สำหรับการตรวจสอบความครบถ้วนขององค์ประกอบของไทมด์เพทริเน็ตที่อยู่ในผลลัพธ์การแปลงอีเวนท์บี มีกรณีทดสอบย่อยดังข้อ 1.4) ถึง 1.7) ต่อไปนี้

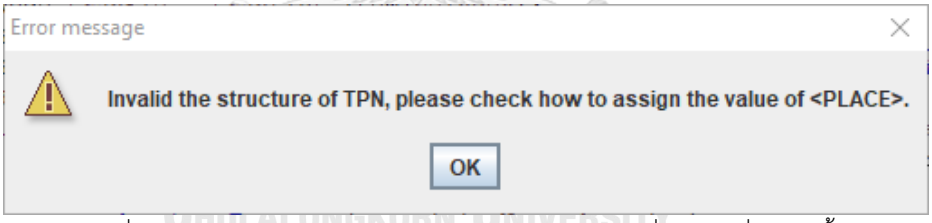
1.4) ตรวจสอบกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำ

กรณีนี้มีการระบุชื่อเพลส p1 เข้ามาซ้ำกัน แสดงในรูปที่ 5.7

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <TPN>
3      <PLACES>
4          <PLACE id="PL_p1" name="p1" marking="2" type="node"/>
5          <PLACE id="PL_p2" name="p1" marking="0" type="node"/>
6      </PLACES>
```

รูปที่ 5.7 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำ

ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำ แสดงดังรูปที่ 5.8



รูปที่ 5.8 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำ

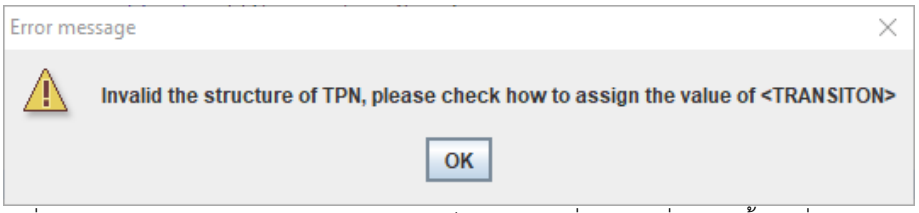
1.5) ตรวจสอบกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำกับชื่อทรานซิชัน

กรณีนี้มีการระบุชื่อเพลสคือ t1 ซึ่งซ้ำกับชื่อของทรานซิชัน t1 แสดงในรูปที่ 5.9

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <TPN>
3      <PLACES>
4          <PLACE id="PL_p1" name="t1" marking="2" type="node"/>
5          <PLACE id="PL_p2" name="p1" marking="0" type="node"/>
6      </PLACES>
7      <TRANSITIONS>
8          <TRANSITION id="TR_t1" name="t1" durationTime="2"  type="node"/>
9          <TRANSITION id="TR_t2" name="t2" durationTime="4"  type="node"/>
10     </TRANSITIONS>
```

รูปที่ 5.9 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำกับชื่อทรานซิชัน

ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำกับชื่อทรานซิชัน แสดงดังรูปที่ 5.10



รูปที่ 5.10 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุชื่อเพลสซ้ำกับชื่อทรานซิชัน

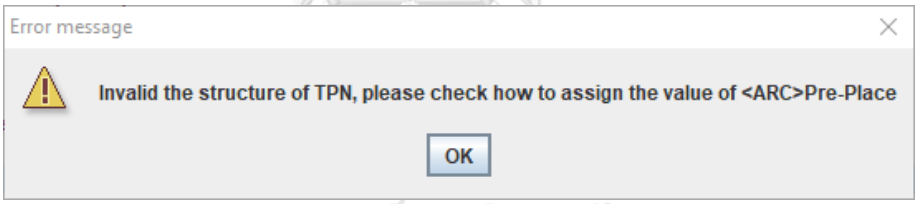
1.6) ตรวจสอบกรณีที่ผู้ใช้ระบุเส้นทางจากทรานซิชันไปทรานซิชัน

กรณีนี้มีการระบุเส้นทางจากทรานซิชันไปยังทรานซิชัน ซึ่งละเมิดนิยามของไทมด์เพทริเน็ต แสดงดังรูปที่ 5.11

```
11      <ARCS>
12          <ARC id="inputPlace" fromNode="t1" toNode="t2" type="node"/>
13          <ARC id="outputPlace" fromNode="t1" toNode="p2" type="node"/>
14      </ARCS>
```

รูปที่ 5.11 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุเส้นทางจากทรานซิชันไปทรานซิชัน

ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุเส้นทางจากทรานซิชันไปทรานซิชัน แสดงดังรูปที่ 5.12



รูปที่ 5.12 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุเส้นทางจากทรานซิชันไปทรานซิชัน

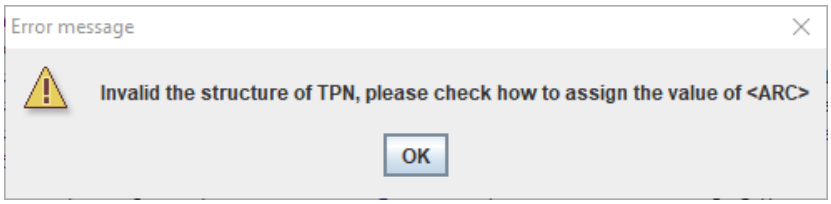
1.7) ตรวจสอบกรณีที่ผู้ใช้ไม่ระบุเส้นทาง

กรณีนี้มีไม่มีการระบุเส้นทางของแบบจำลองไทมด์เพทริเน็ต แสดงดังรูปที่ 5.13

```
7       <TRANSITIONS>
8           <TRANSITION id="TR_t1" name="t1" durationTime="2"  type="node"/>
9           <TRANSITION id="TR_t2" name="t2" durationTime="4"  type="node"/>
10      </TRANSITIONS>
11      <ARCS>
12      </ARCS>
```

รูปที่ 5.13 กรณีทดสอบเพื่อตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุไม่ระบุเส้นทาง

ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุไม่ระบุเส้นทาง แสดงดังรูปที่ 5.14



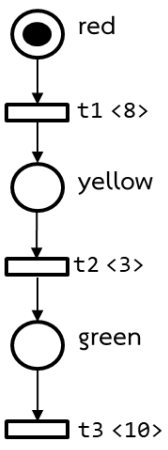
รูปที่ 5.14 ผลการตรวจสอบแฟ้มเอกสารเอกซ์เอ็มแอลกรณีที่ผู้ใช้ระบุไม่ระบุเส้นทาง

5.3.2 ประเมินผลเครื่องมือการแปลงไทมด์เพทริเน็ตไปเป็นอีเวนท์บีโดยอัตโนมัติ โดยใช้กรณีศึกษาสำหรับตรวจสอบคุณสมบัติของผลลัพธ์การแปลงอีเวนท์บีโดยใช้เครื่องมือโรดิน

เมื่อตรวจสอบความถูกต้องของโครงสร้างและไวยากรณ์ของข้อมูลนำเข้าเรียบร้อยแล้ว ผู้วิจัยได้เลือกระบบไฟจราจร, ระบบการทำงานแบบทำงานพร้อมกัน (Concurrent system) และระบบห่วงโซ่อุปทาน (Supply Chain system) ดำเนินการตรวจสอบอีเวนท์บีโดยใช้เครื่องมือโรดิน

5.3.2.1 ทวนสอบการทำ งานของระบบไฟจราจร

เริ่มต้น ผู้วิจัยได้นำแบบจำลองไทมด์เพทริเน็ตสำหรับระบบไฟจราจร ประกอบด้วยเพลส 3 เพลส คือ red, yellowและ green ทรานซิชัน 2 ทรานซิชัน คือ t1, t2 และ t3 และมีมาร์กกิงเริ่มต้นคือ m_0(red) อธิบายการทำงานของระบบจะเริ่มต้นที่เพลส red ทรานซิชัน t1 จะถูกเปิดใช้งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t1 จะถูกยิงเมื่อตัวนับเวลาถึงระยะเวลาที่ 8 โทเค็นจะถูกลบออกและเพิ่มเข้าไปในเพลส yellow ทรานซิชัน t2 จะถูกเปิดใช้งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t2 จะถูกยิงเมื่อตัวนับเวลาถึงระยะเวลาที่ 3 โทเค็นจะถูกลบออกและเพิ่มเข้าไปในเพลส green ทรานซิชัน t3 จะถูกเปิดใช้งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t3 จะถูกยิงเมื่อตัวนับเวลาถึงระยะเวลาที่ 10 แสดงแบบจำลองไทมด์เพทริเน็ตได้ดังรูปที่ 5.15

สัญลักษณ์กราฟิก	ข้อกำหนดโครงสร้างสัญลักษณ์
	TPNs=(P, T, F, V, m_0, D) P={red, yellow, green} T={t1, t2, t3} F ={(red,t_1),(yellow,t_2),(green,t_3),(t_1, yellow), (t_2, green)} V = F {(red,t_1) →1,(yellow,t_2) →1, (green,t_3) →1,(t_1, yellow) →1,(t_2, green) →1} m_0=(1, 0, 0) D=(8, 3, 10)

รูปที่ 5.15 ระบบไฟจราจรที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ต

จากรูปที่ 5.15 ผู้วิจัยได้นำระบบไฟจราจรที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ตไประบุในแฟ้มเอกสารเอกซ์เอ็มแอลที่ได้จัดเตรียมไว้ แสดงดังรูปที่ 5.16

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<TPN>
    <PLACES>
        <PLACE id="PL_red" name="red" marking="1" type="node"/>
        <PLACE id="PL_yellow" name="yellow" marking="0" type="node"/>
        <PLACE id="PL_green" name="green" marking="0" type="node"/>
    </PLACES>
    <TRANSITIONS>
        <TRANSITION id="TR_t1" name="t1" durationTime="8"  type="node"/>
        <TRANSITION id="TR_t2" name="t2" durationTime="3"  type="node"/>
        <TRANSITION id="TR_t3" name="t3" durationTime="10"  type="node"/>
    </TRANSITIONS>
    <ARCS>
        <ARC id="inputPlace" fromNode="red" toNode="t1" type="node"/>
        <ARC id="outputPlace" fromNode="t1" toNode="yellow" type="node"/>
        <ARC id="inputPlace" fromNode="yellow" toNode="t2" type="node"/>
        <ARC id="outputPlace" fromNode="t2" toNode="green" type="green"/>
        <ARC id="inputPlace" fromNode="green" toNode="t3" type="node"/>
    </ARCS>
</TPN>
```

รูปที่ 5.16 แฟ้มเอกสารเอกซ์เอ็มแอลที่ระบุส่วนประกอบของระบบไฟจราจรที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ต

จากนั้นผู้วิจัยได้ดำเนินการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับระบบไฟจราจรให้เป็นอีเวนท์บีโค้ดโดยใช้เครื่องมือการแปลงที่ได้พัฒนามาเรียบร้อยแล้ว แสดงผลลัพธ์อีเวนท์บีโดยแบ่งเป็น 2 ส่วนประกอบคือ ส่วนประกอบริบทแสดงในรูปที่ 5.17 และส่วนประกอบแมชชีนแสดงในรูปที่ 5.18

```
context C_TrafficLight
sets PLACE TRANSITION STATUS
constants red yellow green t1 t2 t3 hold
          enabled firing reset durationTime WEIGHT PREPL POSTPL
axioms
      @axm0 partition(PLACE,{red},{yellow},{green})
      @axm1 partition(TRANSITION,{t1},{t2},{t3})
      @axm2 partition(STATUS,{hold},{enabled},{firing},{reset})
      @axm3 PREPL : TRANSITION<->PLACE
      @axm4 PREPL={t1|->red,t3|->green,t2|->yellow}
      @axm5 POSTPL : TRANSITION<->PLACE
      @axm6 POSTPL={t2|->green,t1|->yellow}
      @axm7 WEIGHT : NAT
      @axm8 WEIGHT=1
      @axm9 durationTime : TRANSITION --> NAT
      @axm10 durationTime= {t3|->10,t2|->3,t1|->8}
end
```

รูปที่ 5.17 ผลลัพธ์การแปลงส่วนประกอบบริบทของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบไฟจราจร

```
machine m_TrafficLight sees c_TrafficLight
variables s_PLACE s_TRANSITION s_MARKING STAT TIK postPlace
invariants
     @inv1 s_PLACE <: PLACE
     @inv2 s_TRANSITION <: TRANSITION
     @inv3 s_MARKING : s_PLACE --> INT
     @inv4 STAT : TRANSITION --> STATUS
     @inv5 TIK : s_TRANSITION --> NAT
     @inv6 postPlace <: s_PLACE
events
event INITIALISATION
then
     @act1 s_PLACE := PLACE
     @act2 s_TRANSITION := TRANSITION
     @act3 s_MARKING := {red|->1,yellow|->0,green|->0}
     @act4 STAT := {t1|->hold,t2|->hold,t3|->hold}
     @act5 TIK := {t1|->0,t2|->0,t3|->0}
     @act6 postPlace := {}
end
event
    …
end
```

รูปที่ 5.18 ผลลัพธ์การแปลงส่วนประกอบแมชชีนของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบไฟจราจร

เมื่อดำเนินการแปลงเรียบร้อยแล้ว ผู้วิจัยได้นำผลลัพธ์การแปลงไปใส่ในเครื่องมือโรดินเพื่อทวนสอบขั้นตอนการทำงานซึ่งผู้วิจัยมีเป้าหมายในการทวนสอบ ดังต่อไปนี้

1) ตรวจสอบคุณสมบัติด้านความปลอดภัยของระบบ (Safety Property)

เป้าหมายการทวนสอบเพื่อตรวจสอบสมบัติด้านความปลอดภัยของระบบคือ ระบบจะไม่มีพบตัวอย่างคาน (Counterexample) โดยเลือกใช้ตรรกศาสตร์เชิงกาลเวลา คือ

■ ตรวจสอบพฤติกรรมถูกเปิดใช้งาน (Enable) คือ เมื่อเพลสใดๆ มีโทเค็นเท่ากันจำนวน WEIGHT แล้วสถานะของทรานซิชันที่อยู่ปลายทางจะอยู่ในสถานะถูกเปิดใช้ STATE(t) เท่ากับ enabled แสดงตรรกศาสตร์เชิงกาลเวลา คือ

G {(s_MARKING(red)=WEIGHT => STAT(t1) = enabled)
& (s_MARKING(yellow)=WEIGHT => STAT(t2) = enabled)
& (s_MARKING(green)=WEIGHT => STAT(t3) = enabled)}

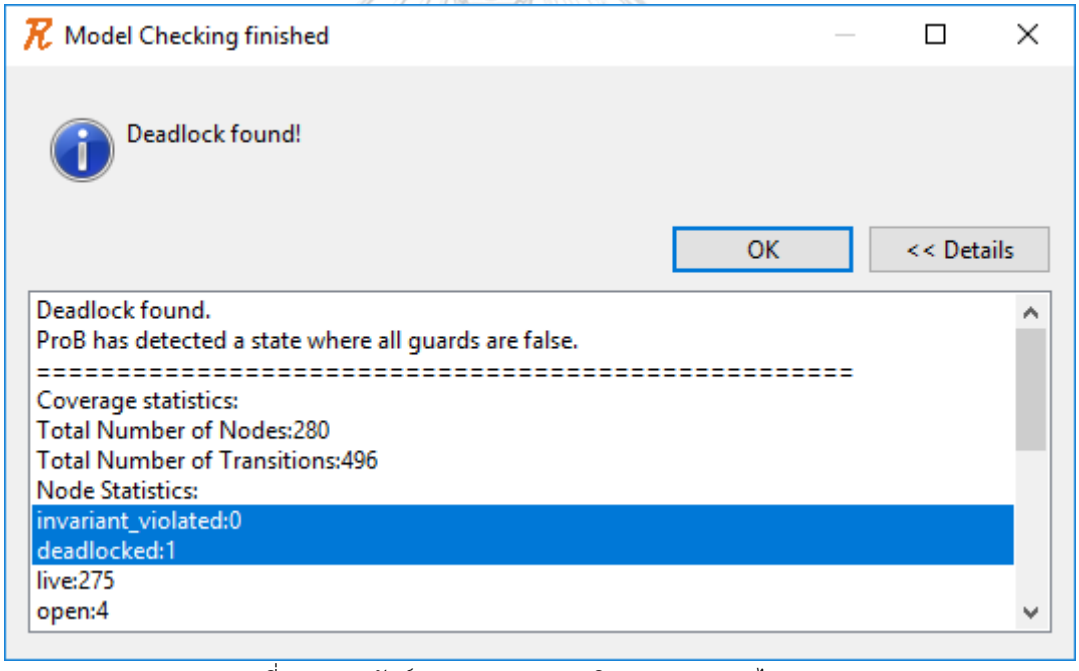
■ ตรวจสอบพฤติกรรมการถูกเปิดใช้งาน (Firing) คือ ทรานซิชันใดๆ t อยู่ในสถานะถูกยิง STAT(t) เท่ากับ firing แล้วจะมีค่าของตัวนับเวลา TIK(t) เท่ากับค่าของระยะเวลาของทรานซิชัน durationTime(t) เสมอ แสดงตรรกศาสตร์เชิงกาลเวลา คือ

$$G\ \{(STAT(t1) = firing \Rightarrow TIK(t1) = durationTime(t1))$$
$$\&\ (STAT(t2) = firing \Rightarrow TIK(t2) = durationTime(t2))$$
$$\&\ (STAT(t3) = firing \Rightarrow TIK(t3) = durationTime(t3))\}$$

จากการทวนสอบจะพบว่าการคุณสมบัติด้านความปลดภัยโดยการตรวจสอบพฤติกรรมถูกเปิดใช้งานและพฤติกรรมเมื่อทรานซิชันถูกยิงของของระบบไฟจราจร จะพบว่าผลลัพธ์การทวนสอบคือไม่พบตัวอย่างค้านเกิดขึ้น ซึ่งเป็นไปตามที่เป้าหมายที่ตั้งไว้

2) ตรวจสอบหาภาวะติดตายของระบบ (Deadlock)

เป้าหมายการทวนสอบเพื่อตรวจสอบหาภาวะติดตายของระบบไฟจราจร คือ เมื่อทวนสอบระบบจะพบภาวะติดตายเกิดขึ้น จากการทวนสอบระบบไฟจราจร สรุปได้ว่าผลลัพธ์ทวนสอบเพื่อหาภาวะติดตายเป็นไปตามเป้าหมายที่ตั้งไว้ แสดงผลลัพธ์การทวนสอบดังรูปที่ 5.19



รูปที่ 5.19 ผลลัพธ์ตรวจสอบหาภาวะติดตายของระบบไฟจราจร

3) ตรวจสอบลำดับการยิงสำหรับแต่ละทรานซิชันในอีเวนท์บีต้องสอดคล้องกับนิยามของไทมด์เพทริเน็ต

เป้าหมายการทวนเพื่อตรวจสอบลำกับการยิงของทรานซิชันของระบบไฟจราจร คือ เมื่อทรานซิชันใดๆ t1 อยู่ในสถานะ firing แล้วตัวนับเวลา TIK(t) จะต้องเท่ากับ durationTime(t) เมื่อทรานซิชัน t1 ถูกยิง ทรานซิชัน t2 จะอยู่ในสถานะ enabled จากนั้นตัวนับเวลาจะเพิ่มขึ้นทีละ 1 เมื่อตัวนับเวลา TIK(t2) มี

ค่าเท่ากับระยะเวลา durationTime(t2) สถานะ t2 จะเปลี่ยนไปเป็น firing จากนั้น ทรานซิชัน t2 จะอยู่ในสถานะ enabled และจะเปลี่ยนไปอยู่ในสถานะ firing เมื่อตัวนับเวลา TIK(t2) มีค่าเท่ากับระยะเวลา durationTime(t2) แสดงตัวได้ลำดับการเกิดได้ในตารางที่ 5.1

ตารางที่ 5.1 เป้าหมายการทวนเพื่อตรวจสอบลำกับการยิงของทรานซิชันของระบบไฟจราจร

ลำดับ	ทรานซิชัน	สถานะ (STAT)	ตัวนับเวลา (TIK)	ระยะเวลา (durationTime)
1	t1	firing	8	8
2	t2	enabled	0	3
3	t2	firing	3	3
4	t3	enabled	0	10
5	t3	firing	10	10

สำหรับการตรวจสอบลำดับการยิงสำหรับแต่ละทรานซิชัน ผู้วิจัยเลือกใช้วิธีการ Simulation model ผลลัพธ์การตรวจสอบเวลาการยิงสำหรับแต่ละทรานซิชัน จากการทวนสอบระบบไฟจราจร สรุปได้ว่าผลลัพธ์ทวนสอบเพื่อหาภาวะติดตายเป็นไปตามเป้าหมายที่ตั้งไว้ แสดงผลลัพธ์การทวนสอบดังรูปที่ 5.20

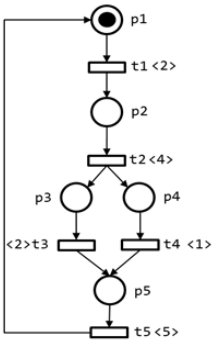
m_TrafficLight	STAT	TIK	durationTime
Fire(green,t3)	{(t1↦hold),(t [#5] (t3↦firing)}	{(t1↦0),(t2↦0),(t3↦10)}	{(t1↦8),(t2↦3),(t3↦10)}
Enabled(green,t3)	{(t1↦hold),(t [#4] t3↦enabled)}	{(t1↦0),(t2↦0),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
resetIndividualTransition(yellow,t2)	{(t1↦hold),(t2↦hold),(t3↦hold)}	{(t1↦0),(t2↦0),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
removeToken(yellow,t2)	{(t1↦hold),(t2↦reset),(t3↦hold)}	{(t1↦0),(t2↦3),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
e_DynamicTransition(yellow,green,t2)	{(t1↦hold),(t2↦firing),(t3↦hold)}	{(t1↦0),(t2↦3),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
Fire(yellow,t2)	{ [#3]),(t2↦firing),(t3↦hold)}	{(t1↦0),(t2↦3),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
Counting(t2)	{(t1↦hold),(t2↦enabled),(t3↦hold)}	{(t1↦0),(t2↦3),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
resetIndividualTransition(red,t1)	{(t1↦hold),(t2↦enabled),(t3↦hold)}	{(t1↦0),(t2↦2),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
Counting(t2)	{(t1↦reset),(t2↦enabled),(t3↦hold)}	{(t1↦8),(t2↦2),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
removeToken(red,t1)	{(t1↦reset),(t2↦enabled),(t3↦hold)}	{(t1↦8),(t2↦1),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
Counting(t2)	{(t1↦firing),(t2↦enabled),(t3↦hold)}	{(t1↦8),(t2↦1),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
Enabled(yellow,t2)	{(t1 [#2] t2↦enabled),(t3↦hold)}	{(t1↦8),(t2↦0),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
e_DynamicTransition(red,yellow,t1)	[#1] (t1↦firing),(t2↦hold),(t3↦hold)}	{(t1↦8),(t2↦0),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}
Fire(red,t1)	(t1↦firing),(t2↦hold),(t3↦hold)}	{(t1↦8),(t2↦0),(t3↦0)}	{(t1↦8),(t2↦3),(t3↦10)}

รูปที่ 5.20 ผลลัพธ์การตรวจสอบเวลาการยิงสำหรับแต่ละทรานซิชัน

5.3.2.2 ทวนสอบการทำงานของระบบการทำงานแบบพร้อมกัน (Concurrent system)

เริ่มต้น ผู้วิจัยได้นำแบบจำลองไทมด์เพทริเน็ตสำหรับระบบการทำงานแบบพร้อมกันประกอบด้วยเพลส 5 เพลส คือ p1, p2, p3, p4 และ p5 ทรานซิชัน 5 ทรานซิชัน คือ t1, t2, t3, t4 และ t5 และมีมาร์กกิงเริ่มต้นคือ $m_0(p1)$ อธิบายการทำงานของระบบจะเริ่มต้นที่เพลส p1 ทรานซิชัน t1 จะถูกเปิดใช้งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t1 จะถูกยิงเมื่อตัวนับเวลาถึงระยะเวลาที่ 2 โทเค็นจะถูกลบออกและเพิ่มเข้าไปในเพลส p2 ทรานซิชัน t2 จะถูกเปิดใช้งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t2 จะถูกยิงเมื่อตัวนับเวลาถึงระยะเวลาที่ 4 โทเค็นจะถูกลบออกและเพิ่มเข้าไปในเพลส p3 และ p4 ทรานซิชัน t3 และ t4 จะถูกเปิดใช้งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t3 cและ t4 จะถูกยิงเมื่อตัวนับเวลาถึงระยะเวลาที่ 2 และ 1 ตามลำดับ โทเค็นจะถูกลบออกและเพิ่มเข้าไปในเพลส p5 ทรานซิชัน t5 จะถูกเปิดใช้

งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t5 จะถูกยิงเมื่อตัวนับเวลาถึงระยะเวลาที่ 5 โทเค็นจะถูกลบออกและเพิ่มเข้าไปในเพลส p1 แสดงแบบจำลองไทมด์เพทริเน็ตได้ดังรูปที่ 5.21

สัญลักษณ์กราฟิก	ข้อกำหนดโครงสร้างสัญลักษณ์
	TPNs=(P, T, F, V, m_0, D) P={p1, p2, p3, p4, p5} T={t1, t2, t3, t4, t5} F ={(p_1,t_1),(p_2,t_2),(p_3,t_3),(p_4,t_4),(p_5,t_5), (t_1,p_2),(t_2,p_3),(t_2,p_4),(t_3,p_5),(t_4,p_5),(t_5,p_1)} V = F {(p_1,t_1) →1,(p_2,t_2) →1,(p_3,t_3) →1, (p_4,t_4) →1, (p_5,t_5) →1,(t_1,p_2) →1, (t_2,p_3) →1,(t_2,p_4) →1, (t_3,p_5) →1, (t_4,p_5) →1,(t_5,p_1) →1} m_0=(1, 0, 0, 0, 0, 0) D=(2, 4, 2, 1, 5)

รูปที่ 5.21 ระบบการทำงานแบบทำงานพร้อมกันที่อยู่ในรูปแบบของจำลองไทมด์เพทริเน็ต

จากรูปที่ 5.21 ผู้วิจัยได้นำระบบไฟจราจรที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ตไประบุในแฟ้มเอกสารเอกซ์เอ็มแอลที่ได้จัดเตรียมไว้ แสดงดัง รูปที่ 5.22

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<TPN>
    <PLACES>
        <PLACE id="PL_p1" name="p1" marking="1" type="node"/>
        <PLACE id="PL_p2" name="p2" marking="0" type="node"/>
        <PLACE id="PL_p3" name="p3" marking="0" type="node"/>
        <PLACE id="PL_p4" name="p4" marking="0" type="node"/>
        <PLACE id="PL_p5" name="p5" marking="0" type="node"/>
    </PLACES>
    <TRANSITIONS>
        <TRANSITION id="TR_t1" name="t1" durationTime="2"  type="node"/>
        <TRANSITION id="TR_t2" name="t2" durationTime="4"  type="node"/>
        <TRANSITION id="TR_t3" name="t3" durationTime="2"  type="node"/>
        <TRANSITION id="TR_t4" name="t4" durationTime="1"  type="node"/>
        <TRANSITION id="TR_t5" name="t5" durationTime="5"  type="node"/>
    </TRANSITIONS>
    <ARCS>
        <ARC id="inputPlace" fromNode="p1" toNode="t1" type="node"/>
        <ARC id="outputPlace" fromNode="t1" toNode="p2" type="node"/>
        <ARC id="inputPlace" fromNode="p2" toNode="t2" type="node"/>
        <ARC id="outputPlace" fromNode="t2" toNode="p3" type="node"/>
        <ARC id="outputPlace" fromNode="t2" toNode="p4" type="node"/>
        <ARC id="inputPlace" fromNode="p3" toNode="t3" type="node"/>
        <ARC id="inputPlace" fromNode="p4" toNode="t4" type="node"/>
        <ARC id="outputPlace" fromNode="t3" toNode="p5" type="node"/>
        <ARC id="outputPlace" fromNode="t4" toNode="p5" type="node"/>
        <ARC id="inputPlace" fromNode="p5" toNode="t5" type="node"/>
        <ARC id="outputPlace" fromNode="t5" toNode="p1" type="node"/>

    </ARCS>
</TPN>
```

รูปที่ 5.22 แฟ้มเอกสารเอกซ์เอ็มแอลที่ระบุส่วนประกอบของระบบการทำงานแบบพร้อมกันที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ต

จากนั้นผู้วิจัยได้ดำเนินการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับระบบการทำงานแบบทำงานพร้อมกันให้เป็นอีเวนท์บีโค้ดโดยใช้เครื่องมือการแปลงที่ได้พัฒนามาเรียบร้อยแล้ว แสดงผลลัพธ์อีเวนท์บีโดยแบ่งเป็น 2 ส่วนประกอบคือ ส่วนประกอบบริบทแสดงในรูปที่ 5.23 และส่วนประกอบแมชชีนแสดงในรูปที่ 5.24

```
context c_Concurrent
sets PLACE TRANSITION STATUS
constants p1 p2 p3 p4 p5 t1 t2 t3 t4 t5 hold
          enabled firing reset durationTime WEIGHT PREPL POSTPL
axioms
      @axm0 partition(PLACE,{p1},{p2},{p3},{p4},{p5})
      @axm1 partition(TRANSITION,{t1},{t2},{t3},{t4},{t5})
      @axm2 partition(STATUS,{hold},{enabled},{firing},{reset})
      @axm3 PREPL ∈ TRANSITION⇔PLACE
      @axm4 PREPL={t5↦p5,t4↦p4,t3↦p3,t2↦p2,t1↦p1}
      @axm5 POSTPL ∈ TRANSITION⇔PLACE
      @axm6 POSTPL={t3↦p5,t2↦p4,t1↦p2,t4↦p5,t5↦p1}
      @axm7 WEIGHT ∈ ℕ
      @axm8 WEIGHT=1
      @axm9 durationTime ∈ TRANSITION → ℕ
      @axm10 durationTime= {t3↦2,t2↦4,t1↦2,t4↦1,t5↦5}
end
```

รูปที่ 5.23 ผลลัพธ์การแปลงส่วนประกอบบริบทของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบการทำงานแบบพร้อมกัน

```
machine m_Concurrent  sees c_Concurrent
variables s_PLACE s_TRANSITION s_MARKING STAT TIK postPlace
invariants
  @inv1 s_PLACE ⊆ PLACE
  @inv2 s_TRANSITION ⊆ TRANSITION
  @inv3 s_MARKING ∈ s_PLACE → ℤ
  @inv4 STAT ∈ TRANSITION → STATUS
  @inv5 TIK ∈ s_TRANSITION → ℕ
  @inv6 postPlace ⊆ s_PLACE
events
  event INITIALISATION
    then
    @act1 s_PLACE ≔ PLACE
    @act2 s_TRANSITION ≔ TRANSITION
    @act3 s_MARKING ≔ {p1↦1,p2↦0,p3↦0,p4↦0,p5↦0}
    @act4 STAT ≔ {t1↦hold,t2↦hold,t3↦hold,t4↦hold,t5↦hold}
    @act5 TIK ≔ {t1↦0,t2↦0,t3↦0,t4↦0,t5↦0}
    @act6 postPlace ≔ ∅
  end
```

รูปที่ 5.24 ผลลัพธ์การแปลงส่วนประกอบแมชชีนของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบการทำงานแบบพร้อมกัน

เมื่อดำเนินการแปลงเรียบร้อยแล้ว ผู้วิจัยได้นำผลลัพธ์การแปลงไปใส่ในเครื่องมือโรดินเพื่อทวนสอบขั้นตอนการทำงาน โดยผู้วิจัยดำเนินการทวนสอบตามกรณีทดสอบ คือ

1) ตรวจสอบคุณสมบัติด้านความปลอดภัยของระบบ (Safety Property)

เป้าหมายการทวนสอบเพื่อตรวจสอบสมบัติด้านความปลอดภัยของระบบการทำงานแบบพร้อมกัน คือ ระบบจะไม่มีพบตัวอย่างคาน (Counterexample) โดยเลือกใช้ตรรกศาสตร์เชิงกาลเวลา คือ

- ตรวจสอบพฤติกรรมถูกเปิดใช้งาน (Enable) คือ เมื่อเพลสใดๆ มีโทเค็นเท่ากันจำนวน WEIGHT แล้วสถานะของทรานซิชันที่อยู่ปลายทางจะอยู่ในสถานะถูกเปิดใช้ STAT(t) เท่ากับ enabled แสดงตรรกศาสตร์เชิงกาลเวลา คือ

G {(s_MARKING(p1)=WEIGHT => STAT(t1) = enabled)
& (s_MARKING(p2)=WEIGHT => STAT(t2) = enabled)
& (s_MARKING(p3)=WEIGHT => STAT(t3) = enabled)
& (s_MARKING(p4)=WEIGHT => STAT(t4) = enabled)
& (s_MARKING(p5)=WEIGHT => STAT(t5) = enabled)}

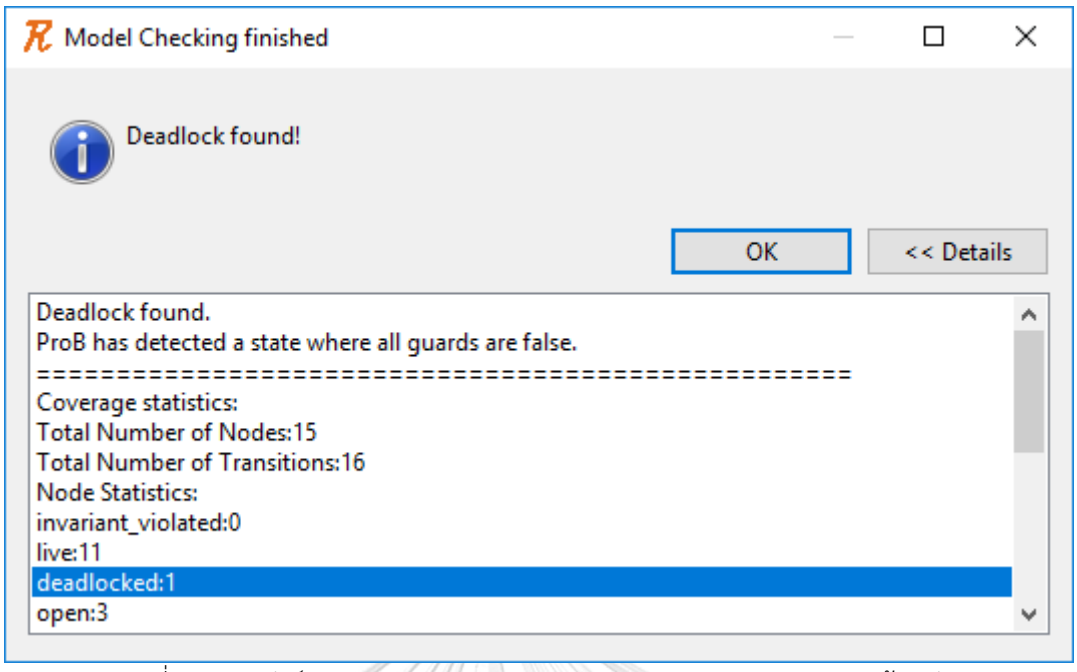
- ตรวจสอบพฤติกรรมการถูกเปิดใช้งาน (Firing) คือ ทรานซิชันใดๆ t อยู่ในสถานะถูกยิง STAT(t) เท่ากับ firing แล้วจะมีค่าของตัวนับเวลา TIK(t) เท่ากับค่าของระยะเวลาของทรานซิชัน durationTime(t) เสมอ แสดงตรรกศาสตร์เชิงกาลเวลา คือ

G {(STAT(t1) = firing => TIK(t1) = durationTime(t1))
& (STAT(t2) = firing => TIK(t2) = durationTime(t2))
& (STAT(t3) = firing => TIK(t3) = durationTime(t3))
& (STAT(t4) = firing => TIK(t4) = durationTime(t4))
& (STAT(t5) = firing => TIK(t5) = durationTime(t5))}

จากการทวนสอบจะพบว่าการคุณสมบัติด้านความปลดภัยโดยการตรวจสอบพฤติกรรมถูกเปิดใช้งานและพฤติกรรมเมื่อทรานซิชันถูกยิงของระบบการทำงานแบบพร้อมกัน จะพบว่าผลลัพธ์การทวนสอบคือไม่พบตัวอย่างค้านเกิดขึ้น ซึ่งเป็นไปตามเป้าหมายที่ตั้งไว้

1) ตรวจสอบหาภาวะติดตายของระบบ (Deadlock)

เป้าหมายการทวนสอบเพื่อตรวจสอบหาภาวะติดตายของระบบการทำงานแบบทำงานพร้อมกัน คือ เมื่อทวนสอบระบบจะพบภาวะติดตายเกิดขึ้น จากการทวนสอบระบบการทำงานแบบพร้อมกัน สรุปได้ว่าผลลัพธ์ทวนสอบเพื่อหาภาวะติดตายเป็นไปตามเป้าหมายที่ตั้งไว้ แสดงผลลัพธ์การทวนสอบดังรูปที่ 5.25



รูปที่ 5.25 ผลลัพธ์ตรวจสอบหาภาวะติดตายของระบบการทำงานแบบทำงานพร้อมกัน

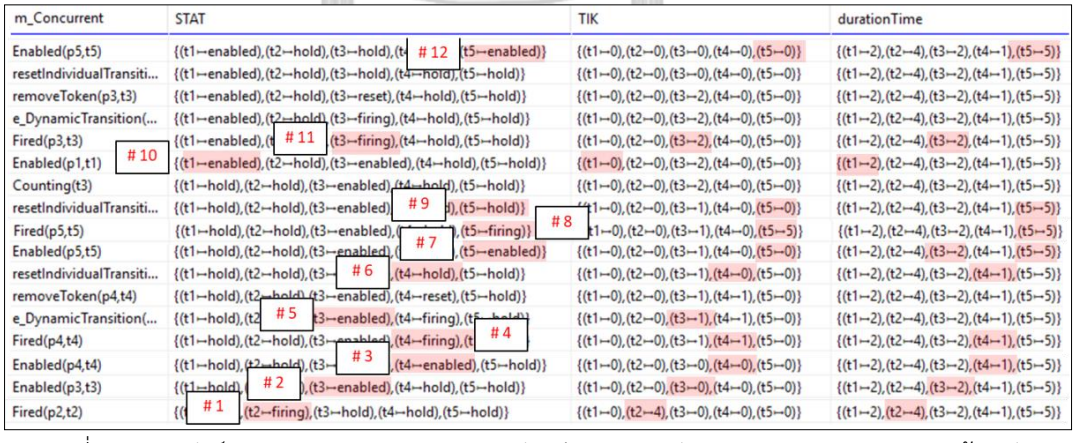
2) ตรวจสอบลำดับการยิงสำหรับแต่ละทรานซิชันในอีเวนท์บีต้องสอดคล้องกับนิยามของไทมด์เพทริเน็ต

เป้าหมายการทวนเพื่อตรวจสอบลำกับการยิงของทรานซิชันของระบบการทำงานแบบทำงานพร้อมกัน คือ เมื่อทรานซิชัน t2 อยู่ในสถานะ firing แล้ว ตัวนับเวลา TIK(t2) จะต้องเท่ากับ durationTime(t2) จากนั้น ทรานซิชัน t3 และ t4 จะอยู่ในสถานะ enabled ถ้า สถานะ t4 เปลี่ยนเป็น firing แล้วสถานะของทรานซิชัน t5 จะเปลี่ยนเป็น enabled เมื่อทรานซิชัน t5 firing แล้ว ทรานซิชัน t1 จะเปลี่ยนสถานะเป็น enable จากนั้น ทรานซิชัน t3 จะเริ่มนับเวลาและสามารถยิงโทเค็นไปยังทรานซิชัน t5 ได้ ตารางที่ 5.2

ตารางที่ 5.2 เป้าหมายการทวนเพื่อตรวจสอบลำกับการยิงของทรานซิชันของระบบการทำงานแบบทำงานพร้อมกัน

ลำดับ	ทรานซิชัน	สถานะ (STAT)	ตัวนับเวลา (TIK)	ระยะเวลา (durationTime)
1	t2	firing	4	4
2	t3	enabled	0	2
3	t4	enabled	0	1
4	t4	firing	1	1
5	t3	enabled	1	2
6	t4	hold	0	1
7	t5	enabled	0	5
8	t5	firing	5	5
9	t5	hold	0	5
10	t1	enabled	0	2
11	t3	firing	2	2
12	t5	enabled	0	5

สำหรับการตรวจสอบลำดับการยิงสำหรับแต่ละทรานซิชัน ผู้วิจัยเลือกใช้วิธีการ Simulation model ผลลัพธ์การตรวจสอบเวลาการยิงสำหรับแต่ละทรานซิชัน จากการทวนสอบระบบการทำงานแบบทำงานพร้อมกัน สรุปได้ว่าผลลัพธ์ทวนสอบเพื่อหาภาวะติดตายเป็นไปตามเป้าหมายที่ตั้งไว้ แสดงผลลัพธ์การทวนสอบดังรูปที่ 5.26



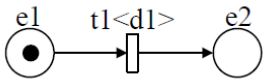
รูปที่ 5.26 ผลลัพธ์การตรวจสอบเวลาการยิงสำหรับแต่ละทรานซิชันของระบบแบบการทำงานพร้อมกัน

5.3.2.3 ทวนสอบการทำงานของระบบห่วงโซ่อุปทาน (Supply Chain)

สำหรับระบบห่วงโซ่อุปทาน ผู้วิจัยได้ศึกษารูปแบบอย่างง่ายของระบบห่วงโซ่อุปทาน [23] และได้นิยามรูปแบบอย่างง่ายออกมาได้ทั้งหมด 4 รูปแบบ ดังต่อไปนี้

1) รูปแบบเหตุและผล (Cause result pattern)

รูปแบบเหตุและผลเป็นรูปแบบอย่างง่าย สำหรับอธิบายความสัมพันธ์ระหว่างสาเหตุของเหตุการณ์ e1 ทำให้เกิดเหตุการณ์ e2 ในระยะเวลาของทรานซิชัน d(t1) แสดงแบบจำลองไทมด์เพทริเน็ตได้ดังรูปที่ 5.27



รูปที่ 5.27 ตัวอย่างของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบของเหตุผล

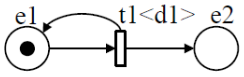
จากรูปที่ 5.27 พบว่ามี 2 เพลสคือ e1 และ e2 และมี 1 ทรานซิชันคือ t1 ที่มีการกำหนดระยะเวลา d1 และมีมาร์กกิงเริ่มต้น ดังนั้นผู้วิจัยจึงได้วิเคราะห์และนำเสนอส่วนประกอบบริบทและแมชชีนของอีเวนท์บีโค้ด แสดงดังรูปที่ 5.28

```
context ct_cause_result_pattern
sets PLACE TRANSITION STATUS
constants e1 e2 t1 durationTime WEIGHT PREPL POSTPL
    axioms
    axm0 partition(PLACE,{e1},{e2})
    axm1 partition(TRANSITION, {t1})
    axm2 PRE = {t1 ↦e1}
    axm3 POST = ∅
    axm4 durationTime = {t1 ↦ d1}
    axm5 WEIGHT =1
end”
“machine OTPN2EBM sees ct_cause_result_pattern
event INITIALISATION
 then
    act0 s_MARKING ≔ {e1 ↦ 1, e2 ↦ 0}
    act1 STAT ≔ {t1↦hold}
    act2 TIK ≔ {t1↦0}
end”
```

รูปที่ 5.28 ผลลัพธ์อีเวนท์บีโค้ดของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบของเหตุผล

2) รูปแบบสาเหตุที่วนกลับ (Repeat cause one effect pattern)

รูปแบบสาเหตุที่วนกลับเป็นรูปแบบ คือการเกิดเหตุการณ์ e1 ภายในระยะเวลาทรานซิชัน d(t1) เป็นสาเหตุให้เกิดเหตุการณ์ e2 และทำให้เกิดเหตุการณ์ e1 ซ้ำอีกครั้ง แสดงแบบจำลองไทมด์เพทริเน็ตได้ดังรูปที่ 5.29



รูปที่ 5.29 ตัวอย่างของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบสาเหตุที่เกิดผลซ้ำ

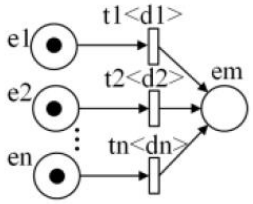
จากรูปที่ 5.29 พบว่ามี 2 เพลส คือ e1 และ e2 และมี 1 ทรานซิชันคือ t1 ที่มีการกำหนดระยะเวลา d1 และมีมาร์กกิงเริ่มต้น ดังนั้นผู้วิจัยจึงได้วิเคราะห์และนำเสนอส่วนประกอบบริบทและแมชชีนของอีเวนท์บีโค้ด แสดงดังรูปที่ 5.30

```
context ct_repeat_cause_one_effect_pattern
sets PLACE TRANSITION STATUS
constants e1 e2 t1 durationTime WEIGHT PREPL POSTPL
axioms
    axm0 partition(PLACE,{e1},{e2})
    axm1 partition(TRANSITION, {t1})
    axm2 PRE = {t1↦e1}
    axm3 PRE = {t1↦e1}
    axm4 POST = {t1↦ e1, t1↦e2}
    axm5 durationTime = {t1 ↦ d1}
    axm6 WEIGHT =1
end”
“machine OTPN2EB sees ct_repeat_cause_one_effect_pattern
event INITIALISATION
  then
    act0 s_MARKING ≔ {e1 ↦ 1, e2 ↦ 0}
    act1 STAT ≔ {t1↦hold}
    act2 TIK ≔ {t1↦0}
end”
```

รูปที่ 5.30 ผลลัพธ์อีเวนท์บีโค้ดของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบสาเหตุที่เกิดผลซ้ำ

3) รูปแบบการเกิดสาเหตุของหลายๆเหตุการณ์ เป็นสาเหตุทำให้เกิดเหตุการณ์หนึ่งเหตุการณ์ (N causes to one result pattern)

รูปแบบการเกิดเหตุการณ์ใดๆ e1, e2,.., en ในระยะเวลาของทรานซิชันใดๆ d(t1), d(t2),...,d(tn) ตามลำดับ ซึ่งเป็นสาเหตุให้เกิดเหตุการณ์ em แสดงแบบจำลองไทมด์เพทริเน็ตได้ดังรูปที่ 5.31



รูปที่ 5.31 ตัวอย่างของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบการเกิดสาเหตุของหลายๆเหตุการณ์ เป็นสาเหตุทำให้เกิดเหตุการณ์หนึ่งเหตุการณ์

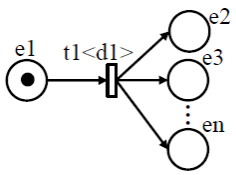
จากรูปที่ 5.31 พบว่ามี 4 เพลส คือ e1, e2, en,..., em และมี 3 ทรานซิชัน คือ t1, t2,..., tn ที่มีการกำหนดระยะเวลา d1, d2,..., dn และมีมาร์กกิงเริ่มต้น ดังนั้นผู้วิจัยจึงได้วิเคราะห์และนำเสนอส่วนประกอบบริบทและแมชชีนของอีเวนท์บีโค้ด แสดงดังรูปที่ 5.32

```
context ct_ N_causes_one_result_pattern
sets PLACE TRANSITION STATUS
constants e1 e2 en em t1 t2 t3 durationTime WEIGHT PREPL
POSTPL
axioms
    axm0 partition(PLACE,{e1},{e2},{en}, …,{em})
    axm1 partition(TRANSITION, {t1}, {t2}, …, {tn})
    axm2 PRE = {t1↦e1, t2↦ e2,…,tn ↦ en}
    axm3 POST = {t1↦ em, t2↦em,…, tn ↦ em}
    axm4 durationTime = {t1 ↦ d1, t2 ↦ d2,…, tn ↦ dn}
    axm5 WEIGHT =1
end
machine OTPN2EBM sees ct_ N_causes_one_result_pattern
event INITIALISATION
  then
    act0 s_MARKING ≔ {e1 ↦ 1, e2 ↦ 1, en ↦ 1,…,em ↦ 0}
    act1 STAT ≔ {t1↦hold, t2↦hold , t3↦hold,…, tn↦hold}
    act2 TIK ≔ {t1↦0, t2↦0, t3↦0,…, tn↦0}
end
```

รูปที่ 5.32 ผลลัพธ์อีเวนท์บีโค้ดของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบการเกิดสาเหตุของหลายๆเหตุการณ์เป็นสาเหตุทำให้เกิดเหตุการณ์หนึ่งเหตุการณ์

4) รูปแบบของหนึ่งเหตุการณ์เป็นสาเหตุทำให้เกิดหลายเหตุการณ์ (One cause to N results pattern)

รูปแบบการเกิดเหตุการณ์ e1 ในระยะเวลาทรานซิชัน d(t1) เป็นสาเหตุให้เกิดเหตุการณ์ใดๆ e2, e3,..,en แสดงแบบจำลองไทมด์เพทริเน็ตได้ดังรูปที่ 5.33



รูปที่ 5.33 ตัวอย่างของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบของหนึ่งเหตุการณ์เป็นสาเหตุทำให้เกิดหลายเหตุการณ์

จากรูปที่ 5.33 พบว่ามี 4 เพลส คือ e1, e2, e3,...,en และมี 3 ทรานซิชัน คือ t1 ที่มีการกำหนดระยะเวลา d1 และมีมาร์กกิงเริ่มต้น ดังนั้นผู้วิจัยจึงได้วิเคราะห์และนำเสนอส่วนประกอบบริบทและแมชชีนของอีเวนท์บีโค้ด แสดงดังรูปที่ 5.34

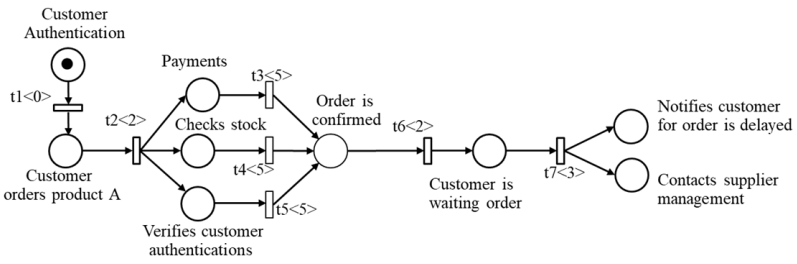
```
context ct_one_cause_N_results_pattern
sets PLACE TRANSITION STATUS
constants e1 e2 e3 en t1 durationTime WEIGHT PREPL POSTPL
axioms
    axm0 partition(PLACE,{e1},{e2},{e3}, …,{en})
    axm1 partition(TRANSITION, {t1})
    axm2 PRE = {t1↦e1}
    axm3 POST = {t1↦ e2, t1↦ e3,…, t1↦ en}
    axm4 durationTime = {t1 ↦ d1}
    axm5 WEIGHT =1
end
machine OTPN2EBM sees ct_one_cause_N_results_pattern
event INITIALISATION
  then
    act0 s_MARKING ≔ {e1 ↦ 1, e2 ↦ 0, e3 ↦ 0, en ↦ 0}
    act1 STAT ≔ {t1↦hold}
    act3 TIK ≔ {t1↦0}
end
```

รูปที่ 5.34 ผลลัพธ์อีเวนท์บีโค้ดของแบบจำลองไทมด์เพทริเน็ตสำหรับรูปแบบของหนึ่งเหตุการณ์เป็นสาเหตุทำให้เกิดหลายเหตุการณ์

เมื่อได้นิยามรูปแบบอย่างง่ายของระบบห่วงโซ่อุปทานแล้ว ผู้วิจัยได้นำรูปแบบทั้ง 4 แบบที่ได้นิยามมารวมกันให้เป็นระบบของห่วงโซ่อุปทานเพื่อสร้างแบบจำลองไทมด์เพทริเน็ต ดังปรากฏในรูปที่ 5.35 จะประกอบด้วยเพลส 9 เพลส คือ CustomerAuthentication, CustomerOrdersProduct, Payments, ChecksStock, VerifiesCustomerAuthentications, OrderIsConfirmed, CustomerIsWaitingOrder, NotifiesCustomerForOrderIsDelayed และ ContactsSupplierManagement ทรานซิชัน 7 ทรานซิชัน คือ t1, t2, t3, t4, t5, t6 และ t7 และมีมาร์กกิงเริ่มต้นคือ m0(CustomerAuthentication) อธิบายการทำงานของระบบจะเริ่มต้นที่เพลส CustomerAuthentication ทรานซิชัน t1 จะถูกเปิดใช้งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t1 จะถูกยิงเมื่อตัวนับเวลาถึงระยะเวลาที่ 0 โทเค็นจะถูกลบออกและเพิ่มเข้าไปในเพลส CustomerOrdersProduct ทรานซิชัน t2 จะถูกเปิดใช้งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t2 จะถูกยิงเมื่อตัวนับเวลาถึงระยะเวลาที่ 2 โทเค็นจะถูกลบออกและเพิ่มเข้าไปในเพลส Payments, ChecksStock และ VerifiesCustomerAuthentications ทรานซิชัน t3, t4 และ t5 จะถูกเปิดใช้งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t3 cและ t4 จะถูกยิงเมื่อตัวนับเวลาถึงระยะที่ 5 และ 1 ตามลำดับ โทเค็นจะถูกลบออกและเพิ่มเข้าไปในเพลส OrderIsConfirmed ทรานซิชัน t6 จะถูกเปิดใช้งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t5 จะถูกยิงเมื่อตัวนับเวลาถึงระยะที่ 2 โทเค็นจะถูกลบออกและเพิ่มเข้าไปใน

เพลส CustomerIsWaitingOrder ทรานซิชัน t7 จะถูกเปิดใช้งานและตัวนับเวลาจะเริ่มทำงาน ทรานซิชัน t7 จะถูกยิงเมื่อตัวนับเวลาถึงระยะที่ 3 โทเค็นจะถูกลบออกและเพิ่มเข้าไปในเพลส NotifiesCustomerForOrderIsDelayed และ CustomerIsWaitingOrder แสดงแบบจำลองไทมด์เพทริเน็ตได้ดังรูปที่ 5.35

สัญลักษณ์กราฟิก



ข้อกำหนดโครงสร้างสัญลักษณ์

TPNs=(P, T, F, V, m_0, D)

P={CustomerAuthentication, CustomerOrdersProduct, Payments, ChecksStock, VerifiesCustomerAuthentications, OrderIsConfirmed, CustomerIsWaitingOrder, NotifiesCustomerForOrderIsDelayed, ContactsSupplierManagement}

T={t1, t2, t3,t4,t5,t6,t7}

F ={(CustomerAuthentication,t_1),(CustomerOrdersProduct,t_2), (Payments,t_3),(ChecksStock,t_4),(VerifiesCustomerAuthentications,t_5) (OrderIsConfirmed,t_6),(CustomerIsWaitingOrder,t_7),(t1, CustomerOrdersProduct), (t2,Payments), (t2,ChecksStock), (t2, VerifiesCustomerAuthentications),(t3, OrderIsConfirmed), (t4, OrderIsConfirmed), (t5, OrderIsConfirmed), (t6, CustomerIsWaitingOrder), (t7, NotifiesCustomerForOrderIsDelayed),(t7, ContactsSupplierManagement)}

V = F {(CustomerAuthentication,t_1) →1,(CustomerOrdersProduct,t_2) →1,(Payments,t3) →1, (ChecksStock,t4) →1,(VerifiesCustomerAuthentications,t5) →1, (OrderIsConfirmed,t6) →1,(CustomerIsWaitingOrder,t7) →1, (t1, CustomerOrdersProduct) →1,(t2,Payments) →1, (t2,ChecksStock) →1, (t2, VerifiesCustomerAuthentications) →1, (t3, OrderIsConfirmed) →1, (t4, OrderIsConfirmed) →1, (t5, OrderIsConfirmed) →1, (t6, CustomerIsWaitingOrder) →1, (t7, NotifiesCustomerForOrderIsDelayed) →1, (t7, ContactsSupplierManagement)→1}

m_0=(1, 0, 0, 0, 0, 0, 0, 0)

D=(0, 2, 5, 5, 5,2,3)

รูปที่ 5.35 ระบบห่วงโซ่อุปทานที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ต

จากรูปที่ 5.35 ผู้วิจัยได้นำระบบห่วงโซ่อุปทานที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ตไประบุในแฟ้มเอกสารเอกซ์เอ็มแอลที่ได้จัดเตรียมไว้ แสดงดังรูปที่ 5.36

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<TPN>
    <PLACES>
        <PLACE id="PL_CustomerAuthentication" name="CustomerAuthentication"
        marking="1" type="node"/>
        <PLACE id="PL_CustomerOrdersProduct" name="CustomerOrdersProduct" marking=
        "0" type="node"/>
        <PLACE id="PL_Payments" name="Payments" marking="0" type="node"/>
        <PLACE id="PL_ChecksStock" name="ChecksStock" marking="0" type="node"/>
        <PLACE id="PL_VerifiesCustomerAuthentications" name=
        "VerifiesCustomerAuthentications" marking="0" type="node"/>
        <PLACE id="PL_OrderIsConfirmed" name="OrderIsConfirmed" marking="0" type=
        "node"/>
        <PLACE id="PL_CustomerIsWaitingOrder" name="CustomerIsWaitingOrder"
        marking="0" type="node"/>
        <PLACE id="PL_NotifiesCustomerForOrderIsDelayed" name=
        "NotifiesCustomerForOrderIsDelayed" marking="0" type="node"/>
        <PLACE id="PL_ContactsSupplierManagement" name=
        "ContactsSupplierManagement" marking="0" type="node"/>
    </PLACES>
    <TRANSITIONS>
        <TRANSITION id="TR_t1" name="t1" durationTime="0"  type="node"/>
        <TRANSITION id="TR_t2" name="t2" durationTime="2"  type="node"/>
        <TRANSITION id="TR_t3" name="t3" durationTime="5"  type="node"/>
        <TRANSITION id="TR_t4" name="t4" durationTime="5"  type="node"/>
        <TRANSITION id="TR_t5" name="t5" durationTime="5"  type="node"/>
        <TRANSITION id="TR_t6" name="t6" durationTime="2"  type="node"/>
        <TRANSITION id="TR_t7" name="t7" durationTime="3"  type="node"/>
    </TRANSITIONS>
    <ARCS>
        <ARC id="inputPlace" fromNode="CustomerAuthentication" toNode="t1" type=
        "node"/>
        <ARC id="outputPlace" fromNode="t1" toNode="CustomerOrdersProduct" type=
        "node"/>
        <ARC id="inputPlace" fromNode="CustomerOrdersProduct" toNode="t2" type=
        "node"/>
        <ARC id="outputPlace" fromNode="t2" toNode="Payments" type="node"/>
        <ARC id="outputPlace" fromNode="t2" toNode="ChecksStock" type="node"/>
        <ARC id="outputPlace" fromNode="t2" toNode=
        "VerifiesCustomerAuthentications" type="node"/>
        <ARC id="inputPlace" fromNode="Payments" toNode="t3" type="node"/>
        <ARC id="outputPlace" fromNode="t3" toNode="OrderIsConfirmed" type="node"/>
        <ARC id="inputPlace" fromNode="ChecksStock" toNode="t4" type="node"/>
        <ARC id="outputPlace" fromNode="t4" toNode="OrderIsConfirmed" type="node"/>
        <ARC id="inputPlace" fromNode="VerifiesCustomerAuthentications" toNode=
        "t5" type="node"/>
        <ARC id="outputPlace" fromNode="t5" toNode="OrderIsConfirmed" type="node"/>
        <ARC id="inputPlace" fromNode="OrderIsConfirmed" toNode="t6" type="node"/>
        <ARC id="outputPlace" fromNode="t6" toNode="CustomerIsWaitingOrder" type=
        "node"/>
        <ARC id="inputPlace" fromNode="CustomerIsWaitingOrder" toNode="t7" type=
        "node"/>
        <ARC id="outputPlace" fromNode="t7" toNode=
        "NotifiesCustomerForOrderIsDelayed" type="node"/>
        <ARC id="outputPlace" fromNode="t7" toNode="ContactsSupplierManagement"
        type="node"/>
    </ARCS>
</TPN>
```

รูปที่ 5.36 แฟ้มเอกสารเอกซ์เอ็มแอลที่ระบุส่วนประกอบของระบบการทำงานแบบพร้อมกันที่อยู่ในรูปแบบของแบบจำลองไทมด์เพทริเน็ต

จากนั้นผู้วิจัยได้ดำเนินการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับห่วงโซ่อุปทานให้เป็นอีเวนท์บีโค้ดโดยใช้เครื่องมือการแปลงที่ได้พัฒนามาเรียบร้อยแล้ว แสดงผลลัพธ์อีเวนท์บีโดยแบ่งเป็น 2 ส่วนประกอบคือ ส่วนประกอบบริบทแสดงใน รูปที่ 5.37 และส่วนประกอบแมชชีนใน รูปที่ 5.38

```
context c_SupplyChain
sets PLACE TRANSITION STATUS
constants CustomerAuthentication CustomerOrdersProduct Payments
      ChecksStock                VerifiesCustomerAuthentications
      OrderIsConfirmed                    CustomerIsWaitingOrder
      NotifiesCustomerForOrderIsDelayed
      ContactsSupplierManagement  t1  t2  t3  t4  t5  t6  t7  hold
      enabled firing reset durationTime WEIGHT PREPL POSTPL
axioms
      @axm0 partition(PLACE,{CustomerAuthentication},
            {CustomerOrdersProduct},{Payments},{ChecksStock},
            {VerifiesCustomerAuthentications},{OrderIsConfirmed}
            ,{CustomerIsWaitingOrder},{NotifiesCustomerForOrderI
            sDelayed},{ContactsSupplierManagement})
      @axm1
      partition(TRANSITION,{t1},{t2},{t3},{t4},{t5},{t6},{t7})
      @axm2 partition(STATUS,{hold},{enabled},{firing},{reset})
      @axm3 PREPL ∈ TRANSITION↔PLACE
      @axm4 PREPL={t4↦ChecksStock,t1↦CustomerAuthentication,
            t6↦OrderIsConfirmed,t5↦VerifiesCustomerAuthenticati
            on,t2↦CustomerOrdersProduct,t3↦Payments,t7↦Custome
            rIsWaitingOrder}
      @axm5 POSTPL ∈ TRANSITION↔PLACE
      @axm6
      POSTPL={t7↦ContactsSupplierManagement,t2↦ChecksStock,
            t5↦OrderIsConfirmed,t7↦NotifiesCustomerForOrderIsDe
            layed,t2↦VerifiesCustomerAuthentications,t1↦Custome
            rOrdersProduct,t2↦Payments,t6↦CustomerIsWaitingOrde
            r}
      @axm7 WEIGHT ∈ ℕ
      @axm8 WEIGHT=1
      @axm9 durationTime ∈ TRANSITION → ℕ
      @axm10                                          durationTime=
      {t3↦5,t2↦2,t1↦0,t4↦5,t5↦5,t6↦2,t7↦3}
end
```

รูปที่ 5.37 ผลลัพธ์การแปลงส่วนประกอบบริบทของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบห่วงโซ่อุปทาน

```
machine m_SupplyChain sees c_SupplyChain
variables s_PLACE s_TRANSITION s_MARKING STAT TIK postPlace
invariants
  @inv1 s_PLACE ⊆ PLACE
  @inv2 s_TRANSITION ⊆ TRANSITION
  @inv3 s_MARKING ∈ s_PLACE → ℤ
  @inv4 STAT ∈ TRANSITION → STATUS
  @inv5 TIK ∈ s_TRANSITION → ℕ
  @inv6 postPlace ⊆ s_PLACE
events
  event INITIALISATION
    then
      @act1 s_PLACE ≔ PLACE
      @act2 s_TRANSITION ≔ TRANSITION
      @act3 s_MARKING ≔ {CustomerAuthentication↦1,
            CustomerOrdersProduct↦0,
            Payments↦0,ChecksStock↦0,
            VerifiesCustomerAuthentications↦0,
            OrderIsConfirmed↦0,CustomerIsWaitingOrder↦0,
            NotifiesCustomerForOrderIsDelayed↦0,
            ContactsSupplierManagement↦0}
      @act4 STAT ≔ {t1↦hold,t2↦hold,t3↦hold,t4↦hold,
            t5↦hold,t6↦hold,t7↦hold}
      @act5 TIK ≔ {t1↦0,t2↦0,t3↦0,t4↦0,t5↦0,t6↦0,t7↦0}
      @act6 postPlace ≔ ∅
  End
event
    …
end
```

รูปที่ 5.38 ผลลัพธ์การแปลงส่วนประกอบแมชชีนของอีเวนท์บีจากการแปลงแบบจำลองไทมด์เพทริเน็ตสำหรับจำลองระบบห่วงโซ่อุปทาน

เมื่อดำเนินการแปลงเรียบร้อยแล้ว ผู้วิจัยได้นำผลลัพธ์การแปลงไปใส่ในเครื่องมือโรดินเพื่อทวนสอบขั้นตอนการทำงาน โดยผู้วิจัยดำเนินการทวนสอบตามกรณีทดสอบ คือ

5) ตรวจสอบคุณสมบัติด้านความปลอดภัยของระบบ (Safety Property)

เป้าหมายการทวนสอบเพื่อตรวจสอบสมบัติด้านความปลอดภัยของระบบห่วงโซ่อุปทาน คือ ระบบจะไม่มีพบตัวอย่างคาน (Counterexample) โดยเลือกใช้ตรรกศาสตร์เชิงกาลเวลา คือ

- ตรวจสอบพฤติกรรมการเปิดใช้งาน (Enable) คือ เมื่อเพลสใดๆ มีโทเค็นเท่ากันจำนวน WEIGHT แล้วสถานะของทรานซิชันที่อยู่ปลายทางจะอยู่ในสถานะถูกเปิดใช้ STAT(t) เท่ากับ enabled แสดงตรรกศาสตร์เชิงกาลเวลา คือ

G{(s_MARKING(CustomerAuthentication)=WEIGHT => STAT(t1) = enabled)

& (s_MARKING(CustomerOrdersProduct)=WEIGHT => STAT(t2) = enabled)

& (s_MARKING(Payments)=WEIGHT => STAT(t3) = enabled)

& (s_MARKING(ChecksStock)=WEIGHT => STAT(t4) = enabled)

& (s_MARKING(VerifiesCustomerAuthentications)=WEIGHT => STAT(t5) = enabled)

& (s_MARKING(OrderIsConfirmed)=WEIGHT => STAT(t6) = enabled)

& (s_MARKING(CustomerIsWaitingOrder)=WEIGHT => STAT(t7) = enabled)}

■ ตรวจสอบพฤติกรรมการถูกเปิดใช้งาน (Firing) คือ ทรานซิชันใดๆ t อยู่ในสถานะถูกยิง STAT(t) เท่ากับ firing แล้วจะมีค่าของตัวนับเวลา TIK(t) เท่ากับค่าของระยะเวลาของทรานซิชัน durationTime(t) เสมอ แสดงตรรกศาสตร์เชิงกาลเวลา คือ

G {(STAT(t1) = firing => TIK(t1) = durationTime(t1))

& (STAT(t2) = firing => TIK(t2) = durationTime(t2))

& (STAT(t3) = firing => TIK(t3) = durationTime(t3))

& (STAT(t4) = firing => TIK(t4) = durationTime(t4))

& (STAT(t5) = firing => TIK(t5) = durationTime(t5))

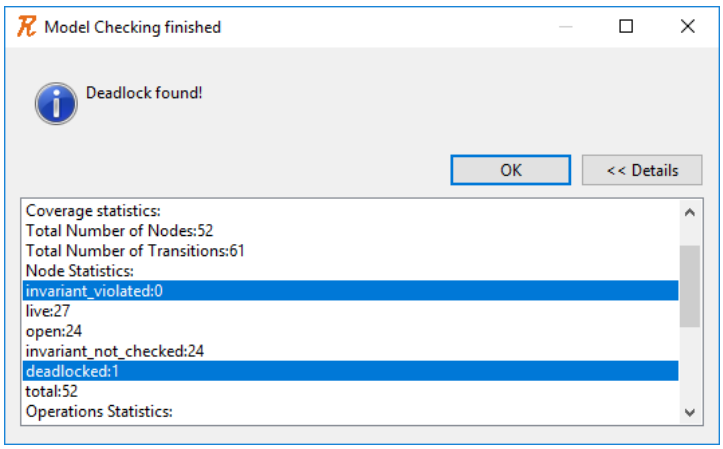
& (STAT(t6) = firing => TIK(t6) = durationTime(t6))

& (STAT(t7) = firing => TIK(t7) = durationTime(t7))}

จากการทวนสอบจะพบว่าการคุณสมบัติด้านความปลดภัยโดยการตรวจสอบพฤติกรรมถูกเปิดใช้งาน และพฤติกรรมเมื่อทรานซิชันถูกยิงของระบบการห่วงโซ่อุปทาน จะพบว่าผลลัพธ์การทวนสอบคือไม่พบตัวอย่างค้านเกิดขึ้น ซึ่งเป็นไปตามเป้าหมายที่ตั้งไว้

6) ตรวจสอบหาภาวะติดตายของระบบ (Deadlock)

เป้าหมายการทวนสอบเพื่อตรวจสอบหาภาวะติดตายของระบบห่วงโซ่อุปทาน คือ เมื่อทวนสอบระบบไม่พบภาวะติดตายเกิดขึ้น จากการทวนสอบระบบห่วงโซ่อุปทาน สรุปได้ว่าผลลัพธ์ทวนสอบเพื่อหาภาวะติดตายเป็นไปตามเป้าหมายที่ตั้งไว้ แสดงผลลัพธ์การทวนสอบดังรูปที่ 5.39



รูปที่ 5.39 ผลลัพธ์ตรวจสอบหาภาวะติดตายของระบบห่วงโซ่อุปทาน

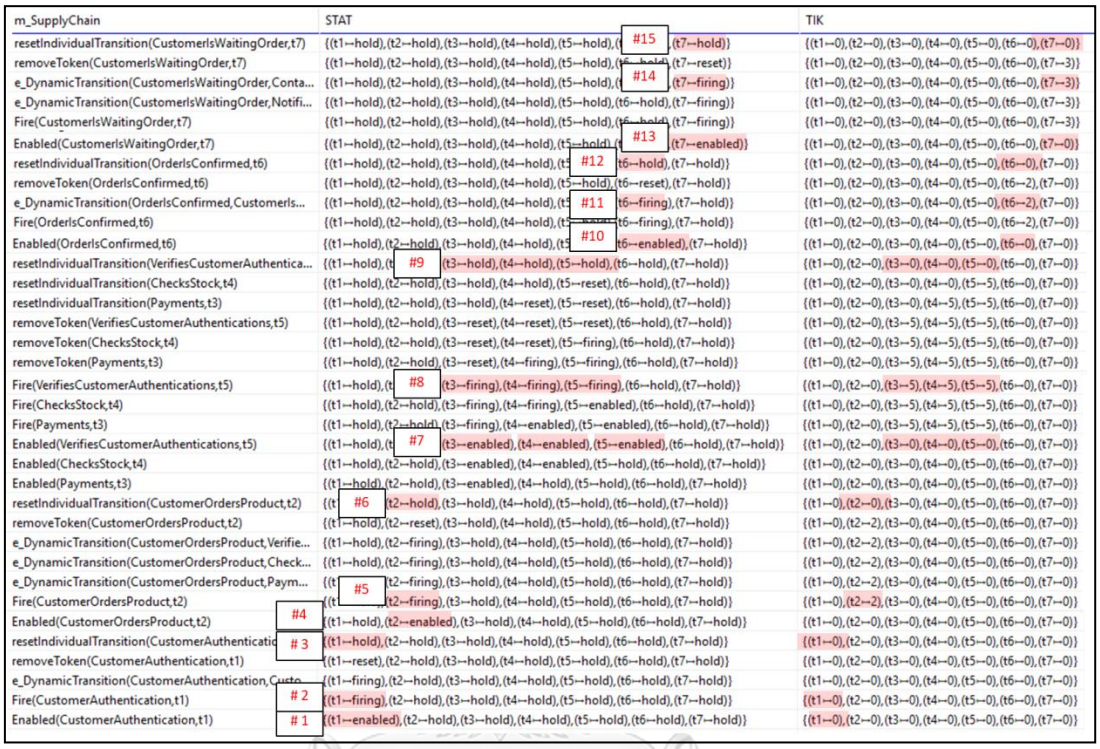
7) ตรวจสอบลำดับการยิงสำหรับแต่ละทรานซิชันในอีเวนท์บีต้องสอดคล้องกับนิยามของไทมด์เพทริเน็ต

เป้าหมายการทวนเพื่อตรวจสอบลำกับการยิงของทรานซิชันของห่วงโซ่อุปทานคือ เมื่อ ตัวนับเวลาของทรานซิชันมีค่าเท่ากับระยะเวลาของทรานซิชันใดๆ ลำดับการเปลี่ยนสถานะของทรานซิชันใดๆ จะเรียงลำดับคือ enabled, firing และ hold ตามลำดับแสดงตัวได้ลำดับการเกิดได้ในตารางที่ 5.3

ตารางที่ 5.3 เป้าหมายการทวนเพื่อตรวจสอบลำกับการยิงของทรานซิชันของระบบห่วงโซ่อุปทาน

ลำดับ	ทรานซิชัน	สถานะ (STAT)	ตัวนับเวลา (TIK)	ระยะเวลา (durationTime)
1	t1	enabled	0	0
2	t1	firing	0	0
3	t1	hold	0	0
4	t2	enabled	0	2
5	t2	firing	2	2
6	t2	hold	2	2
7	t3,t4,t5	enabled	0	5
8	t3,t4,t5	firing	5	5
9	t3,t4,t5	hold	5	5
10	t6	enabled	0	2
11	t6	firing	2	2
12	t6	hold	0	2
13	t7	enabled	0	2
14	t7	firing	3	3
15	t7	hold	0	3

สำหรับการตรวจสอบลำดับการยิงสำหรับแต่ละทรานซิชัน ผู้วิจัยเลือกใช้วิธีการ Simulation model ผลลัพธ์การตรวจสอบเวลาการยิงสำหรับแต่ละทรานซิชัน จากการทวนสอบระบบการทำงานแบบทำงานพร้อมกันสรุปได้ว่าผลลัพธ์ทวนสอบเพื่อหาภาวะติดตายเป็นไปตามเป้าหมายที่ตั้งไว้ แสดงผลลัพธ์การทวนสอบดังรูปที่ 5.40



m_SupplyChain	STAT	TIK
resetIndividualTransition(CustomerIsWaitingOrder,t7)	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
removeToken(CustomerIsWaitingOrder,t7)	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--reset)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--3)}
e_DynamicTransition(CustomerIsWaitingOrder,Conta...	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--firing)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--3)}
e_DynamicTransition(CustomerIsWaitingOrder,Notifi...	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--firing)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--3)}
Fire(CustomerIsWaitingOrder,t7)	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--firing)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--3)}
Enabled(CustomerIsWaitingOrder,t7)	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--enabled)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
resetIndividualTransition(OrderIsConfirmed,t6)	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
removeToken(OrderIsConfirmed,t6)	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--reset),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--2),(t7--0)}
e_DynamicTransition(OrderIsConfirmed,CustomerIs...	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--firing),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--2),(t7--0)}
Fire(OrderIsConfirmed,t6)	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--firing),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--2),(t7--0)}
Enabled(OrderIsConfirmed,t6)	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--enabled),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
resetIndividualTransition(VerifiesCustomerAuthentica...	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
resetIndividualTransition(ChecksStock,t4)	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--reset),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--5),(t6--0),(t7--0)}
resetIndividualTransition(Payments,t3)	{(t1--hold),(t2--hold),(t3--hold),(t4--reset),(t5--reset),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--5),(t5--5),(t6--0),(t7--0)}
removeToken(VerifiesCustomerAuthentications,t5)	{(t1--hold),(t2--hold),(t3--reset),(t4--reset),(t5--reset),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--5),(t4--5),(t5--5),(t6--0),(t7--0)}
removeToken(ChecksStock,t4)	{(t1--hold),(t2--hold),(t3--reset),(t4--reset),(t5--firing),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--5),(t4--5),(t5--5),(t6--0),(t7--0)}
removeToken(Payments,t3)	{(t1--hold),(t2--hold),(t3--reset),(t4--firing),(t5--firing),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--5),(t4--5),(t5--5),(t6--0),(t7--0)}
Fire(VerifiesCustomerAuthentications,t5)	{(t1--hold),(t2--hold),(t3--firing),(t4--firing),(t5--firing),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--5),(t4--5),(t5--5),(t6--0),(t7--0)}
Fire(ChecksStock,t4)	{(t1--hold),(t2--hold),(t3--firing),(t4--firing),(t5--enabled),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--5),(t4--5),(t5--5),(t6--0),(t7--0)}
Fire(Payments,t3)	{(t1--hold),(t2--hold),(t3--firing),(t4--enabled),(t5--enabled),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--5),(t4--5),(t5--5),(t6--0),(t7--0)}
Enabled(VerifiesCustomerAuthentications,t5)	{(t1--hold),(t2--hold),(t3--enabled),(t4--enabled),(t5--enabled),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
Enabled(ChecksStock,t4)	{(t1--hold),(t2--hold),(t3--enabled),(t4--enabled),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
Enabled(Payments,t3)	{(t1--hold),(t2--hold),(t3--enabled),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
resetIndividualTransition(CustomerOrdersProduct,t2)	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
removeToken(CustomerOrdersProduct,t2)	{(t1--hold),(t2--reset),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--2),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
e_DynamicTransition(CustomerOrdersProduct,Verifie...	{(t1--hold),(t2--firing),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--2),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
e_DynamicTransition(CustomerOrdersProduct,Check...	{(t1--hold),(t2--firing),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--2),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
e_DynamicTransition(CustomerOrdersProduct,Paym...	{(t1--hold),(t2--firing),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--2),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
Fire(CustomerOrdersProduct,t2)	{(t1--hold),(t2--firing),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--2),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
Enabled(CustomerOrdersProduct,t2)	{(t1--hold),(t2--enabled),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
resetIndividualTransition(CustomerAuthenticatic	{(t1--hold),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
removeToken(CustomerAuthentication,t1)	{(t1--reset),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
e_DynamicTransition(CustomerAuthentication,Custo...	{(t1--firing),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
Fire(CustomerAuthentication,t1)	{(t1--firing),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}
Enabled(CustomerAuthentication,t1)	{(t1--enabled),(t2--hold),(t3--hold),(t4--hold),(t5--hold),(t6--hold),(t7--hold)}	{(t1--0),(t2--0),(t3--0),(t4--0),(t5--0),(t6--0),(t7--0)}

รูปที่ 5.40 ผลลัพธ์การตรวจสอบเวลาการยิงสำหรับแต่ละทรานซิชันของระบบห่วงโซ่อุปาทาน

บทที่ 6
สรุปผลการดำเนินงานวิจัย

วิทยานิพนธ์ฉบับนี้ทำการออกแบบกฎการแปลงและพัฒนาเครื่องมืออัตโนมัติสำหรับการแปลงแบบจำลองไทมด์เพทริเน็ตไปเป็นอีเวนท์บี สามรถสรุปผลที่ได้จากการวิจัย ข้อจำกัดและข้อเสนอแนะได้ดังต่อไปนี้

6.1 สรุปผลงานวิจัย

จากการวิจัย ได้ออกแบบกฎการแปลงและพัฒนาเครื่องมืออัตโนมัติสำหรับการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีสำหรับการทวนสอบเชิงรูปนัย ผู้วิจัยได้ออกแบบวิธีกาพัฒนาเครื่องมือโดยเริ่มจากการศึกษาเพื่อออกแบบกฎการแปลงและพัฒนาเครื่องมืออัตโนมัติ ซึ่งจะประกอบโดยเริ่มแรกผู้วิจัยได้วิเคราะห์แบบจำลองไทมด์เพทริเน็ตเพื่อนิยามส่วนประกอบไทมด์เพทริเน็ตให้อยู่ในรูปแบบของแฟ้มเอกซ์เอ็มแอลเพื่อให้ได้เค้าร่างไทมด์เพทริเน็ตที่อยู่ในรูปแบบของแฟ้มเอกสารเอกซ์เอ็มแอล จากนั้นผู้วิจัยได้นิยามกฎการแปลงแบบจำลองไทมด์เพทริเน็ตให้เป็นอีเวนท์บี ทั้งหมด 7 ข้อ คือ

กฎข้อที่ 1 กฎการแปลงส่วนประกอบเพลส

กฎข้อที่ 2 กฎการแปลงส่วนประกอบทรานซิชัน

กฎข้อที่ 3 กฎการแปลงส่วนประกอบของการไหลของเส้นทางการย้ายสถานะ

กฎข้อที่ 4 กฎการแปลงช่วงเวลาของแต่ละทรานซิชัน

กฎข้อที่ 5 กฎการแปลงตัวแปร (Variables) และตัวยืนยง (Invariants)

กฎข้อที่ 6 กฎการแปลงเหตุการณ์เริ่มต้น (Initialisation Events)

กฎข้อที่ 7 กฎการแปลงเหตุการณ์แบบไดนามิก (Dynamic Events)

ซึ่งกฎการแปลงทั้งหมดจะเป็นการนิยามกฎแสดงความสัมพันธ์ระหว่างไทมด์เพทริเน็ตและอีเวนท์บี เพื่อใช้สำหรับเครื่องมืออัตโนมัติสำหรับการแปลงไทมด์เพทริเน็ตให้เป็นอีเวนท์บีสำหรับการทวนสอบเชิงรูปนัย

สำหรับการทดสอบกฎการแปลงและเครื่องมือการแปลงแบบจำลองไทมด์เพทริเน็ตไปเป็นอีเวนท์บี ซึ่งจะแบ่งออกเป็น 2 กลุ่ม คือ

1) ประเมินผลการทำงานของเครื่องมือการแปลงโดยใช้กรณีทดสอบ สำหรับการตรวจสอบไวยากรณ์และความหมายของข้อมูลนำเข้าแฟ้มเอกสารเอกซ์เอ็มแอลไทมด์เพทริเน็ต

สำหรับการตรวจสอบการทำงานของเครื่องมือการแปลงโดยใช้กรณีทดสอบ 7 กรณี ซึ่งผลการตรวจสอบเป็นไปตามเป้าหมายที่ตั้งไว้

2) ประเมินผลเครื่องมือการแปลงโดยใช้กรณีศึกษาสำหรับตรวจสอบคุณสมบัติของผลลัพธ์การแปลงอีเวนท์บีโดยใช้เครื่องมือโรดิน

สำหรับการประเมินผลเครื่องมือการแปลงโดยใช้กรณีศึกษา 3 กรณี คือ การทวนสอบระบบไฟจราจร การทวนสอบระบบการทำงานแบบพร้อมกันและระบบห่วงโซ่อุปทาน ซึ่งผลการตรวจสอบเป็นไปตามเป้าหมายที่ตั้ง

เครื่องมืออัตโนมัตินี้จะเป็นทางเลือกสำหรับให้ผู้ใช้งานที่ต้องการทวนสอบระบบโดยใช้วิธีการพิสูจน์ทางคณิตศาสตร์โดยใช้เครื่องมือโรดิน ผู้ใช้สามารถเลือกใช้แบบจำลองไทมด์เพทริเน็ต ซึ่งผู้ใช้สามารถทำความเข้าใจองค์ประกอบได้โดยง่าย โดยที่ผู้ใช้งานไม่จำเป็นต้องมีพื้นฐานทางคณิตศาสตร์ในการเขียนอีเวนท์บีสำหรับการทวนสอบระบบ

6.2 ข้อจำกัดของเครื่องมือการแปลง

1) กฎการแปลงที่นิยามสามารถรองรับไทมด์เพทริเน็ตที่มีค่าน้ำหนักของการเปลี่ยนสถานะเท่ากับ 1 และเพลสถูกจำกัดด้วยจำนวน 1 โทเค็น หากต้องการเพิ่มค่าน้ำหนักของการเปลี่ยนสถานะและจำนวนโทเค็นในเพลสต้องมีการปรับเปลี่ยนกฎที่นิยามไว้

2) เครื่องมือการแปลงสามารถแสดงผลลัพธ์การแปลงในรูปแบบของแฟ้มเอกสารข้อความ ซึ่งการนำไปเปิดใช้ในเครื่องมือโรดิน ผู้ใช้จำเป็นต้องส่วนประกอบในเครื่องมือโรดินและคัดลอกผลลัพธ์การแปลงไปตรวจสอบในเครื่องมือโรดินด้วยตัวเอง

3) ลำดับการยิงโดยใช้ส่วนขยายโปรบียังไม่อัตโนมัติ ผู้ใช้จำเป็นต้องเข้าใจลำดับพฤติกรรมของไทมด์เพทริเน็ต เพื่อเลือกดูลำดับการเกิดเหตุการณ์ที่เกิดขึ้นในขณะนั้น

6.3 ข้อเสนอแนะ

1) ออกแบบกฎการแปลงให้ครอบคลุมค่าน้ำหนักและจำนวนโทเค็นในเพลสที่มีค่ามากกว่า 1 เพื่อสนับสนุนการเปลี่ยนสถานะหนึ่งไปอีกสถานะหนึ่งของไทมด์เพทริเน็ตได้

2) พัฒนาเครื่องมือให้เปิดใช้เครื่องมือโรดินได้อย่างอัตโนมัติ โดยที่ผู้ใช้ไม่จำเป็นต้องคัดลอกผลลัพธ์การแปลงและนำไปวางในเครื่องมือโรดินด้วยตัวเอง

รายการอ้างอิง

[1] M. Heiner and M. Heisel, "Modeling Safety-Critical Systems with Z and Petri Nets," Berlin, Heidelberg, 1999, pp. 361-374: Springer Berlin Heidelberg.

[2] R. A. Kemmerer, "Integrating formal methods into the development process," *IEEE Software,* vol. 7, pp. 37-50, 1990.

[3] B. W. Choi, "Petri net approaches for modeling, controlling, and validating flexible manufacturing systems," 1994.

[4] *Petri Nets Tools Database Quick Overview*. Available: https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html

[5] L. Popova-Zeugmann, *Time and Petri Nets*. Berlin, Germany: Springer, 2013.

[6] M. Werner, L. Popova-Zeugmann, and J. Richling, "A Method to Prove Non-Reachability in Priority Duration Petri Nets," *Fundamenta Informaticae,* pp. 1001-1018, 2004.

[7] J.-R. Abrial, *Modeling in Event-B System and Software Engineer*, 1st ed. New York, NY, USA: Cambridge University Press, 2010.

[8] D. Cansell, D. Méry, and J. Rehm, "Time Constraint Patterns for Event B Development," Berlin, Heidelberg, 2006, pp. 140-154: Springer Berlin Heidelberg.

[9] C. Ramchandani, "Analysis of asynchronous concurrent systems by timed petri nets," Massachusetts Institute of Technology, 1973.

[10] J.-R. Abrial, *The B-book: assigning programs to meanings*. Cambridge University Press, 1996, p. 779.

[11] M. Jastram and P. M. Butler, "Rodin User's Handbook: Covers Rodin v. 2.8," 2014.

[12] D. Déharbe, P. Fontaine, Y. Guyot, and L. Voisin, "SMT solvers for Rodin," in *International Conference on Abstract State Machines, Alloy, B, VDM, and Z*, 2012, pp. 194-207: Springer.

[13] T. Bouton, D. C. B. de Oliveira, D. Déharbe, and P. Fontaine, "veriT: an open, trustable and efficient SMT-solver," in *International Conference on Automated Deduction*, 2009, pp. 151-156: Springer.

[14] L. de Moura and N. Bjørner, "Z3: An Efficient SMT Solver," Berlin, Heidelberg, 2008, pp. 337-340: Springer Berlin Heidelberg.

[15] *EU-project DEPLOY*. Available: http://www.deploy-project.eu

[16] M. Leuschel and M. Butler, "ProB: A Model Checker for B," Berlin, Heidelberg, 2003, pp. 855-874: Springer Berlin Heidelberg.

[17] *Linear-time Temporal Logic*. Available: http://www-step.stanford.edu/tutorial/temporal-logic/temporal-logic.html

[18] C. Attiogbé, "Semantic embedding of petri nets into event-B," *arXiv preprint cs/0510073,* 2005.

[19] M. Garoui, B. Mazigh, and A. Koukam, "The EventB2PN Tool: From Event-B specification to Petri Nets through model transformation," in *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2015, pp. 1-7.

[20] S. Korecko and B. Sobota, *Petri Nets to B-Language Transformation in Software Development*. 2014, pp. 187-206.

[21] G. Ciobanu, T. S. Hoang, and A. Stefanescu, "From TiMo to Event-B: Event-Driven Timed Mobility," in *2014 19th International Conference on Engineering of Complex Computer Systems*, 2014, pp. 1-10.

[22] F. Siavashi, M. Waldén, L. Tsiopoulos, and J. Vain, "Modeling Critical Systems with Timing Constraints in Event-B," *NWPT 2013,* p. 70.

[23] R. Liu, A. Kumar, and W. Van Der Aalst, "A formal modeling approach for supply chain event management," *Decision Support Systems,* vol. 43, no. 3, pp. 761-778, 2007.

ภาคผนวก

ภาคผนวก

1) ผลลัพธ์การแปลงอีเวนท์บีโค้ดที่สมบรูณ์ของแบบจำลองไทมด์เพทริเน็ตสำหรับระบบไฟจราจร

```
//=================== Context component=========================//
context c_TrafficLight
sets PLACE TRANSITION STATUS
constants red yellow green t1 t2 t3 hold enabled firing reset
durationTime WEIGHT PREPL POSTPL
axioms
   @axm0 partition(PLACE,{red},{yellow},{green})
   @axm1 partition(TRANSITION,{t1},{t2},{t3})
   @axm2 partition(STATUS,{hold},{enabled},{firing},{reset})
   @axm3 PREPL ∈ TRANSITION⇸PLACE
   @axm4 PREPL={t1↦red,t3↦green,t2↦yellow}
   @axm5 POSTPL ∈ TRANSITION⇸PLACE
   @axm6 POSTPL={t2↦green,t1↦yellow}
   @axm7 WEIGHT ∈ ℕ
   @axm8 WEIGHT=1
   @axm9 durationTime ∈ TRANSITION → ℕ
   @axm10 durationTime= {t3↦10,t2↦3,t1↦8}
end
//=================== Machine component=========================//
machine m_TrafficLight sees c_TrafficLight
variables s_PLACE s_TRANSITION s_MARKING STAT TIK postPlace
invariants
  @inv1 s_PLACE ⊆ PLACE
  @inv2 s_TRANSITION ⊆ TRANSITION
  @inv3 s_MARKING ∈ s_PLACE → ℤ
  @inv4 STAT ∈ TRANSITION → STATUS
  @inv5 TIK ∈ s_TRANSITION → ℕ
  @inv6 postPlace ⊆ s_PLACE
events
  event INITIALISATION
    then
      @act1 s_PLACE ≔ PLACE
      @act2 s_TRANSITION ≔ TRANSITION
      @act3 s_MARKING ≔ {red↦1,yellow↦0,green↦0}
      @act4 STAT ≔ {t1↦hold,t2↦hold,t3↦hold}
      @act5 TIK ≔ {t1↦0,t2↦0,t3↦0}
      @act6 postPlace ≔ ∅
  end
   event Enabled
    any p t
    where
      @grd1 (t ∈ s_TRANSITION ∧ (p∈ PREPL[{t}]) ∧ (STAT(t) = hold) )
```

```
      @grd2 (p ∈ dom(s_MARKING)) ∧ (s_MARKING(p) ≥ WEIGHT) ∧
(postPlace=∅)
    then
      @act1 STAT(t) ≔ enabled
  end
  event Counting
    any t
    where
      @grd1 (t ∈ s_TRANSITION) ∧ (STAT(t) = enabled)
      @grd2 (TIK(t) < durationTime(t))
    then
      @act1 TIK(t) ≔ TIK(t) +1
  end
  event Fire
    any t p
    where
      @grd1 (t∈s_TRANSITION) ∧ (TIK(t) = durationTime(t)) ∧ (STAT(t)
= enabled)
      @grd2 (p ∈ s_PLACE) ∧ (p ∈ PREPL[{t}]) ∧ (POSTPL[{t}] ⊆
s_PLACE)
    then
      @act1 STAT(t) ≔ firing
      @act2 postPlace ≔ postPlace∪ POSTPL[{t}]
  end
  event resetIndividualTransition
    any t p
    where
      @grd1 (t∈ dom(TIK)) ∧ (STAT(t) = reset) ∧ (postPlace = ∅) ∧
(POSTPL[{t}]⊆s_PLACE)
      @grd2 (p ∈ s_PLACE) ∧ (p ∈ PREPL[{t}])
    then
      @act1 STAT(t) ≔ hold
      @act2 TIK(t) ≔ 0
  end
  event resetConcurrentTransition
    any t pIn pOut
    where
      @grd1 (t∈s_TRANSITION) ∧ (STAT(t) = enabled) ∧  (pOut
∈POSTPL[{t}])
      @grd2 (pOut ∈ s_PLACE) ∧ (s_MARKING(pOut) ≠ 0)
      @grd3 (pIn ∈ s_PLACE) ∧ (pIn ∈PREPL[{t}])
    then
      @act1 STAT(t) ≔ hold
      @act2 TIK(t) ≔ 0
      @act3 s_MARKING(pIn) ≔ 0
  end
  event removeToken
    any t pIn
```

```
    where
      @grd1 (pIn ∈ PREPL[{t}]) ∧ (POSTPL[{t}] ⊆ s_PLACE) ∧ (postPlace
= ∅)
      @grd2 (pIn ∈ s_PLACE) ∧ (s_MARKING(pIn) > 0) ∧(STAT(t) =
firing)
    then
      @act1 s_MARKING(pIn) ≔  s_MARKING(pIn) – WEIGHT
      @act2 STAT(t) ≔ reset
  end
  event e_DynamicTransition
    any t pIn pOut
    where
      @grd1 (t∈s_TRANSITION) ∧ (pOut ∈ s_PLACE) ∧ (STAT(t) = firing)
∧ ( pOut ∈ POSTPL[{t}])
      @grd2 (postPlace ≠ ∅) ∧ (pOut ∈ postPlace)
      @grd3 (pIn ∈ s_PLACE) ∧ (pIn ∈ PREPL[{t}])
    then
      @act1 s_MARKING(pOut) ≔  s_MARKING(pOut) + WEIGHT
      @act2 postPlace ≔  postPlace \ {pOut}
  end
end
```

2) ผลลัพธ์การแปลงอีเวนท์บีโค้ดที่สมบูรณ์ของแบบจำลองไทมด์เพทริเน็ตสำหรับระบบการทำงานแบบพร้อมกัน

```
//=================== Context component=========================//
context c_Concurrent
sets PLACE TRANSITION STATUS
constants p1 p2 p3 p4 p5 t1 t2 t3 t4 t5 hold enabled firing reset
durationTime WEIGHT PREPL POSTPL
axioms
  @axm0 partition(PLACE,{p1},{p2},{p3},{p4},{p5})
  @axm1 partition(TRANSITION,{t1},{t2},{t3},{t4},{t5})
  @axm2 partition(STATUS,{hold},{enabled},{firing},{reset})
  @axm3 PREPL ∈ TRANSITION↔PLACE
  @axm4 PREPL={t5↦p5,t4↦p4,t3↦p3,t2↦p2,t1↦p1}
  @axm5 POSTPL ∈ TRANSITION↔PLACE
  @axm6 POSTPL={t3↦p5,t2↦p4,t2↦p3,t1↦p2,t4↦p5,t5↦p1}
  @axm7 WEIGHT ∈ ℕ
  @axm8 WEIGHT=1
  @axm9 durationTime ∈ TRANSITION → ℕ
  @axm10 durationTime= {t3↦2,t2↦4,t1↦2,t4↦1,t5↦5}
end
```

```
//=================== Machine component=========================//
machine m_Concurrent  sees c_Concurrent
variables s_PLACE s_TRANSITION s_MARKING STAT TIK postPlace
invariants
  @inv1 s_PLACE ⊆ PLACE
  theorem @inv2 s_TRANSITION ⊆ TRANSITION
  @inv3 s_MARKING ∈ s_PLACE → ℤ
  @inv4 STAT ∈ TRANSITION → STATUS
  @inv5 TIK ∈ s_TRANSITION → ℕ
  @inv6 postPlace ⊆ s_PLACE
events
  event INITIALISATION
    then
      @act1 s_PLACE ≔ PLACE
      @act2 s_TRANSITION ≔ TRANSITION
      @act3 s_MARKING ≔ {p1↦1,p2↦0,p3↦0,p4↦0,p5↦0}
      @act4 STAT ≔ {t1↦hold,t2↦hold,t3↦hold,t4↦hold,t5↦hold}
      @act5 TIK ≔ {t1↦0,t2↦0,t3↦0,t4↦0,t5↦0}
      @act6 postPlace ≔ ∅
  end
  event Enabled
    any p t
    where
      @grd1 (t ∈ s_TRANSITION ∧ (p∈ PREPL[{t}]) ∧ (STAT(t) = hold) )
      @grd2 (p ∈ dom(s_MARKING)) ∧ (s_MARKING(p) ≥ WEIGHT) ∧ (
postPlace = ∅)
    then
      @act1 STAT(t) ≔ enabled
  end
  event Counting
    any t
    where
      @grd1 (t ∈ s_TRANSITION) ∧ (STAT(t) = enabled)
      @grd2 (TIK(t) < durationTime(t))
    then
      @act1 TIK(t) ≔ TIK(t) +1
  end
  event Fire
    any t p
    where
      @grd1 (t∈s_TRANSITION) ∧ (TIK(t) = durationTime(t)) ∧ (STAT(t)
= enabled)
      @grd2 (p ∈ s_PLACE) ∧ (p ∈ PREPL[{t}]) ∧ (POSTPL[{t}] ⊆
s_PLACE)
    then
      @act1 STAT(t) ≔ firing
      @act2 postPlace ≔ postPlace∪ POSTPL[{t}]
  end
```

```
  event resetIndividualTransition
    any t p
    where
      @grd1 (t∈ dom(TIK)) ∧ (STAT(t) = reset) ∧ (postPlace = ∅) ∧
(POSTPL[{t}]⊆s_PLACE)
      @grd2 (p ∈ s_PLACE) ∧ (p ∈ PREPL[{t}])
    then
      @act1 STAT(t) ≔ hold
      @act2 TIK(t) ≔ 0
  end
  event resetConcurrentTransition
    any t pIn pOut
    where
      @grd1 (t∈s_TRANSITION) ∧ (STAT(t) = enabled) ∧  (pOut
∈POSTPL[{t}])
      @grd2 (pOut ∈ s_PLACE) ∧ (s_MARKING(pOut) ≠ 0)
      @grd3 (pIn ∈ s_PLACE) ∧ (pIn ∈PREPL[{t}])
    then
      @act1 STAT(t) ≔ hold
      @act2 TIK(t) ≔ 0
      @act3 s_MARKING(pIn) ≔ 0
  end
  event removeToken
    any t pIn
    where
      @grd1 (pIn ∈ PREPL[{t}]) ∧ (POSTPL[{t}] ⊆ s_PLACE) ∧ (postPlace
= ∅)
      @grd2 (pIn ∈ s_PLACE) ∧ (s_MARKING(pIn) > 0) ∧(STAT(t) =
firing)
    then
      @act1 s_MARKING(pIn) ≔  s_MARKING(pIn) − WEIGHT
      @act2 STAT(t) ≔ reset
  end
  event e_DynamicTransition
    any t pIn pOut
    where
      @grd1 (t∈s_TRANSITION) ∧ (pOut ∈ s_PLACE) ∧ (STAT(t) = firing)
∧ ( pOut ∈ POSTPL[{t}])
      @grd2 (postPlace ≠ ∅) ∧ (pOut ∈ postPlace)
      @grd3 (pIn ∈ s_PLACE) ∧ (pIn ∈ PREPL[{t}])
    then
      @act1 s_MARKING(pOut) ≔  s_MARKING(pOut) + WEIGHT
      @act2 postPlace ≔  postPlace \ {pOut}
  end
end
```

3) ผลลัพธ์การแปลงอีเวนท์บีโค้ดที่สมบูรณ์ของแบบจำลองไทมด์เพทริเน็ตสำหรับระบบห่วงโซ่อุปทาน

```
//=================== Context component=========================//
context c_SupplyChain
sets PLACE TRANSITION STATUS
constants CustomerAuthentication CustomerOrdersProduct Payments
ChecksStock VerifiesCustomerAuthentications OrderIsConfirmed
CustomerIsWaitingOrder NotifiesCustomerForOrderIsDelayed
ContactsSupplierManagement t1 t2 t3 t4 t5 t6 t7 hold enabled firing
reset durationTime WEIGHT PREPL POSTPL
axioms
  @axm0
partition(PLACE,{CustomerAuthentication},{CustomerOrdersProduct},{Paym
ents},{ChecksStock},{VerifiesCustomerAuthentications},{OrderIsConfirme
d},{CustomerIsWaitingOrder},{NotifiesCustomerForOrderIsDelayed},{Conta
ctsSupplierManagement})
  @axm1 partition(TRANSITION,{t1},{t2},{t3},{t4},{t5},{t6},{t7})
  @axm2 partition(STATUS,{hold},{enabled},{firing},{reset})
  @axm3 PREPL ∈ TRANSITION↔PLACE
  @axm4
PREPL={t4↦ChecksStock,t1↦CustomerAuthentication,t6↦OrderIsConfirmed,
t5↦VerifiesCustomerAuthentications,t2↦CustomerOrdersProduct,t3↦Payme
nts,t7↦CustomerIsWaitingOrder}
  @axm5 POSTPL ∈ TRANSITION↔PLACE
  @axm6
POSTPL={t7↦ContactsSupplierManagement,t2↦ChecksStock,t5↦OrderIsConfi
rmed,t7↦NotifiesCustomerForOrderIsDelayed,t2↦VerifiesCustomerAuthent
ications,t1↦CustomerOrdersProduct,t2↦Payments,t6↦CustomerIsWaitingOr
der}
  @axm7 WEIGHT ∈ ℕ
  @axm8 WEIGHT=1
  @axm9 durationTime ∈ TRANSITION → ℕ
  @axm10 durationTime= {t3↦5,t2↦2,t1↦0,t4↦5,t5↦5,t6↦2,t7↦3}
end
//=================== Machine component=========================//
machine m_SupplyChain sees c_SupplyChain
variables s_PLACE s_TRANSITION s_MARKING STAT TIK postPlace
invariants
  @inv1 s_PLACE ⊆ PLACE
  @inv2 s_TRANSITION ⊆ TRANSITION
  @inv3 s_MARKING ∈ s_PLACE → ℤ
  @inv4 STAT ∈ TRANSITION → STATUS
  @inv5 TIK ∈ s_TRANSITION → ℕ
  @inv6 postPlace ⊆ s_PLACE

events
  event INITIALISATION
    then
```

```
      @act1 s_PLACE ≔ PLACE
      @act2 s_TRANSITION ≔ TRANSITION
      @act3 s_MARKING ≔
{CustomerAuthentication↦1,CustomerOrdersProduct↦0,Payments↦0,ChecksS
tock↦0,VerifiesCustomerAuthentications↦0,OrderIsConfirmed↦0,Customer
IsWaitingOrder↦0,NotifiesCustomerForOrderIsDelayed↦0,ContactsSupplie
rManagement↦0}
      @act4 STAT ≔
{t1↦hold,t2↦hold,t3↦hold,t4↦hold,t5↦hold,t6↦hold,t7↦hold}
      @act5 TIK ≔ {t1↦0,t2↦0,t3↦0,t4↦0,t5↦0,t6↦0,t7↦0}
      @act6 postPlace ≔ ∅
  end
  event Enabled
    any p t
    where
      @grd1 (t ∈ s_TRANSITION ∧ (p∈ PREPL[{t}]) ∧ (STAT(t) = hold) )
      @grd2 (p ∈ dom(s_MARKING)) ∧ (s_MARKING(p) ≥ WEIGHT) ∧ (
postPlace = ∅)
    then
      @act1 STAT(t) ≔ enabled
  end
  event Counting
    any t
    where
      @grd1 (t ∈ s_TRANSITION) ∧ (STAT(t) = enabled)
      @grd2 (TIK(t) < durationTime(t))
    then
      @act1 TIK(t) ≔ TIK(t) +1
  end
  event Fire
    any t p
    where
      @grd1 (t∈s_TRANSITION) ∧ (TIK(t) = durationTime(t)) ∧ (STAT(t)
= enabled)
      @grd2 (p ∈ s_PLACE) ∧ (p ∈ PREPL[{t}]) ∧ (POSTPL[{t}] ⊆
s_PLACE)
    then
      @act1 STAT(t) ≔ firing
      @act2 postPlace ≔ postPlace∪ POSTPL[{t}]
  end
  event resetIndividualTransition
    any t p
    where
      @grd1 (t∈ dom(TIK)) ∧ (STAT(t) = reset) ∧ (postPlace = ∅) ∧
(POSTPL[{t}]⊆s_PLACE)
      @grd2 (p ∈ s_PLACE) ∧ (p ∈ PREPL[{t}])
    then
      @act1 STAT(t) ≔ hold
```

```
      @act2 TIK(t) ≔ 0
  end
  event resetConcurrentTransition
    any t pIn pOut
    where
      @grd1 (t∈s_TRANSITION) ∧ (STAT(t) = enabled) ∧  (pOut
∈POSTPL[{t}])
      @grd2 (pOut ∈ s_PLACE) ∧ (s_MARKING(pOut) ≠ 0)
      @grd3 (pIn ∈ s_PLACE) ∧ (pIn ∈PREPL[{t}])
    then
      @act1 STAT(t) ≔ hold
      @act2 TIK(t) ≔ 0
      @act3 s_MARKING(pIn) ≔ 0
  end
  event removeToken
    any t pIn
    where
      @grd1 (pIn ∈ PREPL[{t}]) ∧ (POSTPL[{t}] ⊆ s_PLACE) ∧ (postPlace
= ∅)
      @grd2 (pIn ∈ s_PLACE) ∧ (s_MARKING(pIn) > 0) ∧(STAT(t) =
firing)
    then
      @act1 s_MARKING(pIn) ≔  s_MARKING(pIn) − WEIGHT
      @act2 STAT(t) ≔ reset
  end
  event e_DynamicTransition
    any t pIn pOut
    where
      @grd1 (t∈s_TRANSITION) ∧ (pOut ∈ s_PLACE) ∧ (STAT(t) = firing)
∧ ( pOut ∈ POSTPL[{t}])
      @grd2 (postPlace ≠ ∅) ∧ (pOut ∈ postPlace)
      @grd3 (pIn ∈ s_PLACE) ∧ (pIn ∈ PREPL[{t}])
    then
      @act1 s_MARKING(pOut) ≔  s_MARKING(pOut) + WEIGHT
      @act2 postPlace ≔  postPlace \ {pOut}
  end
end
```

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวชลิกา ศักดิ์สุภาวัฒนกุล เกิดเมื่อวันที่ 4 ตุลาคม พ.ศ. 2534 ที่จังหวัดขอนแก่น สำเร็จการศึกษาหลักสูตรวิทยาศาสตร์บัณฑิต (วท.บ.) สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2557 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2559